Universidad Carlos III de Madrid

Departamento de Economía de la Empresa

Management Issues in Open Source Software Networks.

Tesis para el doctorado en Economía de la Empresa y Métodos Cuantitativos.

Autor:
Rebeca del Carmen Méndez Durón

Supervisors

Clara E. García
Marco Giarratana

2008

Agradecimientos

Este trabajo no sería posible sin la curiosa sucesión de acontecimientos en mi vida y sin el apoyo de mi familia durante mi carrera profesional.

Primeramente, tengo unos padres increíbles – Aída y Alberto-, que no sólo me han dejado manejar mi carrera profesional a mi gusto y conveniencia, sino que lo han promovido. En su sabiduría parental, sin duda estaba la certeza de que en ocasiones cometería errores, pero que eso me permitiría aprender para cometer más aciertos en el futuro.

Segundo, mis hermanos y mis sobrinos. Mis hermanos, que cada vez que hablo con ellos o los visito me dan un abrazo que me ayude a seguir con mi camino y mis sobrinos, que el deseo por conocerlos y verlos crecer hace que me esfuerce más para poder dedicarles tiempo.

Gracias Aídas, Albertos y Anulfos.

Tercero, mi "hermana mayor", Clara. Una de las personas que más ha influido en mi carrera profesional, al principio indirectamente, pero a partir del 2001, directamente es Clara Eugenia García García. Aún recuerdo su voz, el día 5 de septiembre de 2000, cuando me comentaba que el Master en Análisis y Gestión de la Ciencia y la Tecnología se incorporaba como especialidad al Máster en Economía Industrial. Que las perspectivas de estar año y medio en un máster se extendían a dos años en España. En ese momento, nadie imaginó que continuaría por seis años más, y que ahora siga mi carrera profesional por estas tierras. ¡Los más osados vaticinaron cuatro años! Además, ella me ha guiado en muchas circunstancias y situaciones de todo tipo, lo que ha estrechado estos lazos de casi hermandad.

Cuarto, mi familia española, los Lemus Torres. La vida sería tan diferente si no tuviera el apoyo de Ana, Jesús, Toñi y Carlos. Me han adoptado, y yo soy feliz por ello.

Quinto, mi amiga-hermana Gabriela. Que no obstante que viva en Munich, tenemos una estrecha relación y desde ahí me ha apoyado para terminar.

Debo también agradecer a Marco, que taciturna y tímida –por su forma de ser, pero directamente ha colaborado en la realización de este proyecto; y a Olav, por haber aceptado que fuera a Toronto, sin apenas conocerme. Olav y Constança, me sirvieron también de "paño de lágrimas" cuando la situación en el apartamento se volvía demasiado loca.

No quiero olvidar a Mayra, Neil, Lorena y Ben, quienes han sido grandes amigos. Y deseo recordar a mis compañeros Liliana, Pascual, Kremena, Augusto, Szabi, Peter D. y Peter J. Maud, José, Rómulo, Sara, Avgyro, Sergio, Carlos. Gracias también a Mick.

Gracias a la familia que me espera.

Introduction.

Despite the relevance of large firms' contribution to R&D and innovation, last decades changes in economic activity have come mainly from small firms that interrelate knowledge and abilities to go into business (Holmstrom, 1989). These two contrasting models present features that have been largely studied. The big and vertically integrated firms consolidate all the activities involved in either the productive line or the value chain in centralized structures (Williamson, 2002). These structures concentrate power and decision taking processes in top tier levels, leaving to lower level employees scarce freedom to perform their activities (Faldetta, 2002). Firms that interrelate activities among each other are also known as network based firms, because their production process and outcomes turn up by their interaction within a network. These networks are formed by a number of SMEs, and their advantages have been studied in terms of complementarities, knowledge and learning synergies that foster innovation capabilities (Saxenian, 1994). These two models share the similarity that they are goal oriented, they are intended to produce and market goods in an efficient way either integrating activities or signing alliances with partners.

In recent times, scientific research has devoted attention to open source software industry (Lerner and Tirole, 2001; von Hippel and von Krogh, 2003; Bonaccorsi e Rossi, 2003, among many other), characterized by an asynchronous network of skilled programmers that share software and their algorithms to perform a specialized function. It can be defined as a proto-model of organization that is also based in interrelations but among individuals. These interrelations constitute the main form to accumulate expertise through contribution of network members, but their purpose is essentially different. Professionals develop a software program whose algorithm or code is available for everyone. Software and code can be either freely downloadable or users need to pay a fee for them. In any case, the code procurement allows other programmers to understand how software is made, and to build up new features or fix programming flaws.

Although the knowledge base is the same, open source software industry seems different from software industry. Firms in software industry develop software programs inside their vertically integrated structures. Through integration they can assure their products meet upstream and downstream standards' compatibility requirements (Richardson, 1997). Users only can have access to software through a licensing fee, but never get access to the code.

Internet represents one of the main drivers of open source software increasing relevance (Healy, 2002). Apache server, an open source code product, supports most of Internet transactions. Many of the web pages hosted are also open source code. Moreover, the increased flexibility in programming of open source products has expanded the skilled users base, resembling a threat to vertically integrated software companies. These changes have leaded big firms start to produce open source code products (Parloff, 2004).

Oppositely, open source programmers are facing the opportunity to supply these products to non-expert consumers in the mass market. Unskilled consumers need to have friendlier versions that perfectly fit into their hardware. Thus, programmers started to constitute firms that provide complementary services to their actual open code solutions. These two contrary movements give the impression of convergence in an intermediate set up, that deserves to be analyzed through organizational and economic theories.

From an organizational perspective, the focus of this research is not on networks of firms, whose cooperative agreements act as substitutes of either market based relationships or ownership coordination. In contrast, its main focus is relationships different to social and contractual ties, which exist among individuals jointly developing an uncertain product. How individuals take decisions to allocate resources to product development. How their contributions promote the evolution of knowledge base and strengthen the core capabilities of the virtual organization. How environmental, market and economic conditions sow the ground to the emergence of a new organizational sub-field that should interact with large firms and single individuals interests. And how intellectual property rights may constitute themselves a strategy to exert control over innovation and contribute to the model sustainability.

The present work issues introduced above tested in a network of Open Source Software. In Section I, I present a study on the direction of knowledge exchanges how affects the success of a project. Section III, presents a study about the determinants for contribution in open source projects, where I assess for the effects of the main theoretical explanations and suggests some more. In Section IV, I introduce a study about the effects of quality of productive assets as the determinants for organizational reputation. Section V concludes with final comments.

Returns from Social Capital in Open Source Software Networks

Forthcoming Journal of Evolutionary Economics

Rebeca Méndez-Durón,

Universidad Carlos III de Madrid, Departamento de Economía de la Empresa, C/Madrid 126, Getafe, Madrid, Spain. 28903. E-mail: rebeca.mendez@uc3m.es

Clara E. García

Universidad Carlos III de Madrid, Departamento de Economía de la Empresa, C/Madrid 126, Getafe, Madrid, Spain. 28903. E-mail: claragar@emp.uc3m.es

Open Source Software projects base their operation on collaborative structure for knowledge exchange in the form of provision or reception of information, expertise and feedback on the creation of source code. Here, we address the direction of these knowledge flows among projects throughout social networks and their impact on project success. We identify the roles of membership or contribution that individuals play within projects. We found that connections through contributors who bring their knowledge inbound the project, improve project success; connections through members, who transfer their knowledge towards other projects, enhance project success. Finally, we found that ties through shared membership and contribution hamper project success. The analysis of knowledge flows and their impact on project success imply a translation of returns from investment in social capital, where investment takes the shape of knowledge flows and the returns mean the projects' diffusion over the network.

Keywords: Knowledge, Networks, Social Capital, Software Industry.
JEL: A14, L14, L86, O32, Z13

Research funded by the *Spanish Ministry of Science and Education* grant SEJ-2005-06655/ECON on "Challenges and opportunities of new organizational forms: Lessons from Open Source Software projects". We thank Marco Giarratana, Pablo Ruiz-Verdu, two anonymous journal referees, and EMAEE 2007 conference participants for their comments on earlier versions of this paper.

Correspondence to: R. Méndez-Durón

1. Introduction

The Open Source Software (OSS) movement has recently received an enormous amount of attention as the OSS development process, product commercialisation, and diffusion differ dramatically from those of proprietary software solutions. Individual programmers contribute to the development of core tasks, the debugging, and the improvement of programs until they are finished for different reasons (Raymond, 1999): First, self-interested behaviour – programmers need the software and its improvements for their own purposes- and, alternatively, expectations from reciprocity (Baldwin and Clark, 2006; Bergquist and Ljungberg, 2001; DiBona *et al*., 1999; Kollock, 1994); Second, reputation, and associated rents gained by those who make high quality contributions (Lerner and Tirole, 2001 and 2002;-Lakhani and Wolf, 2005); Third, philanthropic behaviour linked to the enjoyment of the contribution itself (Raymond, 1999).

Current research on OSS focuses on the mechanisms and foundations underlying the network formation since they represent a valuable setting for understanding basic aspects of social organisation, such as cooperation, and the emergence of open networks as enduring forms of governance (Lakhani and Von Hippel, 2003). In OSS networks, cooperation takes place among a large, heterogeneous and ever changing number of developers working simultaneously on different projects where, typically, participants are strangers and, there are no long-term commitments[[1]]. However, without active contribution and participation, collaborative knowledge will not succeed. The OSS gains prominence as projects, independently of their domain and scale, organised according to the principles of a *community of practice* (Brown and Duguid, 1991): open and self-organising networks in which shared values, norms and beliefs rest upon generalised exchange mechanisms and expectations of reciprocity. Then, *pure generalised exchange* might emerge in the presence of altruism, a collective sense of fairness that allows for unilateral giving while preventing free riding behaviour (Ekeh, 1974), and whenever gift givers may choose the recipients (*'fairness-based selective giving'*) (Takahashi, 2000) [[2]]. This strategy materialises as a strong reason why individuals freely reveal innovation-related information within innovation communities (Franke and Shah, 2002).

[1] Programmers are geographically distant; they come from different cultures, languages, traditions and differ in personal, professional and social features.
[2] The fairness-based selective strategy is characterized by actors endowed with options for leaving the current relation and forming a new relation […] but it cannot really explain generalized exchange patterns when actors are strangers (Takahashi, 2000).

From an empirical perspective, at least three questions stand out: First, why individuals participate in generalised exchange mechanisms in the absence of hierarchical structures and rules [3]: a refined understanding of the mechanisms governing exchange relationships between community members (fairness, trust, generalised and reciprocal exchange); Second, the relative importance of skill level versus other relationships in determining an individual position in the network; Third, the differences in the emergence of collaborative structure types and their relation to the success of open source projects.

This paper addresses how differences in collaborative structures influence open source projects' success using a panel data of 2,962 software game projects from SourceForge.net. Our data allow us to identify flows of knowledge and information among projects and the collaborative structures emerging based on software developers' contributions working on those projects.

The architecture of network ties and of interpersonal relations will differ among projects as a result of the projects' heterogeneity in both human and social capital. Programmer skills and knowledge refer to human capital while the capabilities to access resources embedded in their social and professional networks are termed as social capital. Programmers working across different projects may have a wide network of relationships favouring knowledge and information flows and affecting the projects' differential success rates. Previous research on social capital and networks indicates that *strong ties,* while providing relational benefits, are less likely to provide non-redundant knowledge. Moreover, projects requiring the reception of mostly explicit knowledge, as is the case of OSS, benefit from having a network of *weak ties* (Granovetter, 1973; Hansen, 1999) while avoiding the costs of developing trustworthy collaborations, or by exploiting access to unique resources (Burt, 1992). However, open source projects may also benefit from knowledge sharing outside their boundaries, as previous research has shown in the case of work groups (Ancona and Caldwell, 1992). In such a context, work groups' structural diversity contributes to strong effects on performance associated with knowledge sharing. A structurally diverse work group is one in which the members, by virtue of their different organisational affiliations, roles, or positions, can expose the group to unique sources of knowledge (Cummings, 2004).

The OSS projects' collaborative structure is based on knowledge sharing defined here as the provision or reception of information, expertise, and feedback regarding the development

[3] Unilateral resource giving within social and economic exchanges may emerge because of: (1) pure altruistic behaviour; (2) collective norms that punish any form of free riding, and (3) rational choice under game theoretical frameworks in which the existence of incentive structures to solve social dilemmas predominates (Olson, 1965).

and/or modification of a source code. Consequently, open source knowledge sharing represents a generalised exchange supported by the expectation that if a community member provides assistance today (*provision*), someone else will provide him with assistance when he needs it (*reception*). However, research on generalised exchange is more concerned about free-riding issues and individual behaviour/incentives than the structural characteristics of work groups and its effects on success.

Our analysis focuses on the effects of knowledge sharing on projects performance accounting from the effects of both knowledge/information reception and provision. Consequently, the research questions addressed are: (1) Does a focal project that receives knowledge from other actors improves its performance? (2) Are these effects comparable to those obtained from providing knowledge to other actors? In addition (3) what are the effects on performance of a focal project where its members engage in generalised exchange towards other projects? To answer those questions we base ourselves on social capital theories and on groups' structural diversity.

In this paper we consider social capital as the ties between programmers that provide access to information and, consequently, between projects on which they work. However, there is empirical evidence to suggest that the nature of ties explains additional variance in outcomes beyond that explained by structure alone (Lin 1999). While research on social capital and social networks has emphasised the *strong/weak* nature of those ties, we account for the directionality of knowledge flows –inwards and outwards- with respect to a focal project as a structural dimension of diversity since they relate to the role performed by individual programmers in different projects.

We use the project as the unit of analysis; each project represents an event and programmers or actors relate to each other through events, and events relate to other events because of common actors. Though the data analyzed represents a two-mode affiliation networks (Waserman and Faust, 1994; Faust, 1997), our theoretical analysis and modelling depart from previous works; we project a two-mode network into a one-mode event by event network (Sorenson and Stuart, 2008). The modelling of events/projects collaborative structure also introduces some attributes of the actors/programmers on whom the ties form.

Our analysis indicates that knowledge flows among projects are asymmetric as result of both members and contributors' role. We found that connections through *contributors* who bring their programming expertise *inbound* the project, besides being a source of new knowledge, improve project success. Additionally, our findings reveal that connections through members who sign up in other projects hinder the *focal* project success, unless they carry on programming activities transferring their knowledge towards other projects. Finally, we found that ties through shared membership and contribution hamper project success.

This paper presents our analysis as follows. First, we introduce the theoretical framework. In section 3, we present the data and variables used, and in section 4, we show the main results. Finally, we conclude by discussing the findings, limitations and further research implications.

2. Theoretical framework and research hypotheses

Previous studies on large OSS projects such as Linux and Apache support the presence of social capital (Bergquist and Ljunberg, 2001; McKelvey, 2001): as big projects nurture themselves from individual contributions, thus the overall effect is straight forward as soon as they capture new contributors. More recently, empirical evidence suggests that social capital, measured by the density of ties, positively influences collective activity and productivity (Long, 2006). When referring to OSS networks, Zhang (2007) identifies previous ties among a group of programmers as a powerful predictor of further members' affiliation to specific projects. The existence and density of prior ties between the initiator of the project and developers positively influence the probability of a project to attract more individuals. Consequently, in an analogous idea to Putnam's (1995), in OSS networks social capital seems likely to be a substitute for the absence of organisational structures and hierarchical authority devoted to promoting communication, coordination, resource pooling and knowledge sharing, and contributing to improving efficiency and performance.

External knowledge in the open network and the inflow of ideas comes through programmers involved in two or more projects. While programmers differ in their skills, they also differ in their access to resources embedded in their social and professional networks, and those differences are critical in the success of OSS projects.

Social capital entails the underlying structure for exchange, where relations among individuals create and support a common understanding that promotes the generation of a public good exploitable by all individuals within the structure (Coleman, 1988). Social capital is rooted in ties among agents that exchange information and provide access to resources leading to a major emphasis on the density of connections and centrality (Ahuja, 2000; Zajac and Westphal, 1996).

Research on social networks has an established and long tradition in social disciplines because of both interest in patterns of exchange and the important implications of networks in the spread of information and knowledge (Burt, 1992; Granovetter, 1973; Larson, 1992; Powell, 1990; Zaheer and Bell, 2005; Zajac and Westphal, 1996).

Scholars interpret social capital as a metaphor about the gains individuals or groups may obtain by belonging and interacting in social network structures. The result is an increasing and dense body of literature that uses social capital as an independent variable to explain a wide range of social phenomena concluding that individuals or firms interacting within a network perform better than when they are alone (Burt, 2000). The structural dimension of social capital emphasises some network properties including the strength of ties (Granovetter, 1973), structural holes (Burt, 1992) and embeddedness (Gulati and Gargiulo, 1999). Direct ties define strong interactions among projects –throughout their programmers- representing channels for the transfer of useful knowledge (Uzzi, 1996 and 1997; Levin and Cross, 2004). The definition of indirect or weak ties establishes that distant and infrequent interactions play an instrumental role on the diffusion of ideas and information (Granovetter, 1985; Uzzi and Lancaster, 2003). Hence, prior studies found that knowledge sharing positively relates to factors such as strong [intra-group] ties (Wellman and Wortley 1990), co-location (Allen 1977), demographic similarity (Pelled 1996), status similarity (Cohen and Zhou 1991), and a history of prior relationships (Krackhardt 1992).

These ties differ in their effects on knowledge sharing; direct/strong ties based on trust enhance the transfer of both tacit and explicit knowledge while indirect/weak ties allow the transfer of codified knowledge more efficiently (Hansen, 1999; Ahuja, 2000). Previous research indicates that *strong ties,* while providing relational benefits, are less likely to provide non-redundant knowledge. Moreover, projects requiring the receipt of mostly explicit knowledge, as is the case of OSS, benefit from having a network of *weak ties* (Granovetter, 1973; Hansen, 1999) while avoiding the costs of developing trustworthy collaborations, or by exploiting access to unique resources (Burt, 1992).

Social capital benefits are the return to investment strategies oriented to institutionalising group relations into a social network. Portes (1998) considers a *socio-relational* dimension of social capital where agents gain access to resources, and a *stock* dimension comprising the quantity and quality of those resources. OSS projects reflect a common stock of knowledge.

Software projects in general and OSS projects in particular, exhibit at least two properties: (1) they have a modular architecture and (2) outcomes may vary along the design process allowing the introduction of new modules and creating an option value for development. These two properties suggest that projects themselves become a stock of social capital supplied and nurtured by members and contributors.

This paper deals with the value of [external] *knowledge sharing* and its effects on OSS projects' success. Knowledge sharing is defined here as the provision or reception of information, expertise, and feedback regarding the development and/or modification of a source code. Thus, we expect that flows of knowledge incoming to a focal project will contribute to its stock of social capital and its success.

HYPOTHESIS 1. *Knowledge sharing will positively influence a focal project's success while the effects of reception will be stronger than the effects of provision.*

However, instead of focusing on the strength of ties - the *closeness* and *frequency* of interactions in a dyadic relationship- and structural embeddedness (Grewal *et al*., 2006) we focus on the groups' structural diversity. Cummings (2004) has hypothesised that external knowledge sharing will be more strongly associated with performance when work groups are more structurally diverse. A structurally diverse work group is one in which the members, depending on their different organisational affiliations, roles, or positions, can expose the group to unique sources of knowledge. Structural diversity within OSS projects refers to differential access to external sources of knowledge by virtue of programmers' affiliations and roles. Although structural diversity may also relate to previous collaborative experiences and expertise, we do not deal with this issue here.

Thus, we expect OSS projects' members to have a differential impact on project success accordingly to their role, function and their social network. For the purpose of this paper, we label the team workforce appointed by the project leader as *members* of the project; these members of the project may perform different activities, from project administrator to language translator. We denote all persons who contribute to software development as *contributors;* though not all *members* of the project are *contributors* and not all *contributors* are *members* of the project. However, some of them are *members* and *contributors* at the same time.

Since knowledge sharing includes the provision and the reception of information, the effects will be stronger if programmers gain unique insight on both projects (i.e. contributor-contributor). Then,

HYPOTHESIS 2. *External knowledge sharing among projects through individuals contributing to software development in more than one project, will positively influence [a focal] project success.*

Projects with individuals representing different functions (i.e. contributors or members) can access diverse social networks they have established in their respective domains and have an effect on a focal project's success from the perspective of social capital. Therefore, we do not expect those gains to differ when individuals in a focal project represent a dual role (contributor/member).

HYPOTHESIS 3. *External knowledge sharing among projects through members contributing to software development in other projects will positively influence the focal project success independently of the type of role they play in the focal project.*

Woolcock (1998) classified ties as bridging, bonding, and linking social capital. Bridging refers to relations between more distant and heterogeneous members; bonding indicates higher levels of homogeneity, while linking relates to the ability to leverage resources and information beyond the community. We identify 'bonding' ties among OSS projects when members contribute to more than one project since they imply higher levels of knowledge homogeneity. On the other hand, we consider that the relationships between a focal project member/contributor who is solely a member in other projects represent 'bridging' ties. Knowledge homogeneity will result in lower effects on project performance because of a lack of access to new knowledge resources, and

HYPOTHESIS 4. *External knowledge sharing among projects through members contributing to software development in the focal project and other projects will have a lower effect on the focal project's success than when members contributing to the focal project are sole members of other projects.*

The analysis of interaction among projects becomes relevant as the emergence of the OSS movement leads to the creation of repositories for software development. Repositories usually are internet-based sites that provide an organisational infrastructure to allow asynchronous exchanges among programmers, mobilise contribution of people otherwise hard to reach, and screen the contributions from redundant knowledge. A repository integrates a community that benefits from the exchange of knowledge among its members.

If repositories track all artifacts and records, we can identify who are contributors and who are just members of the project. Moreover, we can identify if an individual who is a member of a particular project is simultaneously a member of another project. Thus, depending on the affiliation and the role played by individuals somehow involved in the development in each project, the social network for knowledge exchange within the repository comprises nine forms of ties among each pair of projects: 'member-member', 'contributor-member', 'member-contributor', 'contributor-contributor', 'duality-member', 'duality-contributor', 'member-duality', 'contributor-duality' and 'duality-duality'.

The suitability of OSS repositories for identifying the structural dimension of interactions comes from the structure that a repository provides through knowledge exchange. OSS programmers interact by contributing their knowledge to their own and others' projects but they do not necessarily receive the same amount or quality of contributions from other network actors: yet they generate expectations about others' behaviour for providing their knowledge. Therefore, it is interesting to analyze the knowledge flows throughout the community and their communication paths that sustain and reinforce social capital.

The present study contributes to social capital theory and its empirical analysis in several ways. First, we claim there are different individual strategies for investing in social capital, and these strategies have a relation to structural diversity (role and affiliation). Secondly, we support the fact that social capital influences project's success only if *contributors* provide new valuable knowledge whether inside or outside the *focal* project.

3. Data and analysis

The analysis of social capital in OSS repositories requires a thorough database that on one hand, provides intricate details of project formation, organisation and operation, and on the other hand, provides details about the implication of individuals in the software development.

Furthermore, the analysis itself requires disentangling the relationships occurring through the social network.

3.1. Data setting

The data we use in this analysis come from the SourceForge.net Research Data (Department of Computer Science and Engineering, University of Notre Dame). SourceForge.net is the largest repository of open source software; it hosts over 140,000 projects and gathers over 1.5 million registered users. SourceForge.net belongs to OSTG, Inc, which has shared activity data with the University of Notre Dame for the purpose of academic research on OSS, under the condition of sharing the data further with other researchers interested in open source software phenomenon (Christley and Madey, 2005).

Sourceforge.net as a repository has several characteristics that promote network exchanges. The purpose of the repository is to provide a platform for software development over a worldwide web infrastructure for knowledge exchange. They host projects and provide tools that allow asynchronous communication, stock people's contribution, and screen software for redundant knowledge. To host a project, an initiator should register in the network; and post a message to the platform indicating the type of software, its purpose and target public, the intellectual property regime, the programming language, the user interface, the phase of development and the team workforce that will be responsible for developing and controlling the software. Then Sourceforge.net administrators authorise the creation of a space for hosting the project if it satisfies the main premises for the repository. Once the repository hosts the project, then it registers and controls all its activity. Every time there is a movement in the project, the repository electronically archives the information; these movements may include communications among members, forum posts, and more importantly, the *artifacts* –or modules of software code, produced by the people engaged in programming activities. As an important feature of a repository is to open access to projects to everybody on the worldwide web, every single person, registered or not on Sourceforge.net, may see the project, look at the code, and contribute. The repository records changes in the team workforce, so it is possible to know if there are new members or if some of them have already quitted the project. As repository tracks all artifacts records, we can identify who are *contributors*, who are *members*, and who play the *duality* at each project. Hence, we can observe how projects connect within the repository through individuals. These individuals establish a tie among the available pool of knowledge for each project.

Particularly, our dataset is a sub-sample of 2,962 valid observations over twelve months of projects aimed at developing game software. To get this sample, we look at the monthly dumps of data and select all projects that belong to the category of GNU Public License (GPL) [[4]]. We restrict the sample to such a license to guarantee that projects were not subject to any restriction for copying, adoption and distribution, so all software is equally prone to reach the same audience. Then, we limit the sample to games in a general category. This selection serves the purpose of tapping an appropriate measure of project success. Although we will further discuss performance and success measures, we should advance that we are interested in measuring performance as *market penetration of software*; thus, we should pick a category of projects whose target markets were not constrained by the programming skills and abilities of end users. Furthermore, we restrict the sample for the projects that were alive during the whole sample; we take this decision because dataset classifies a project as 'alive' –when the repository still hosts the project and 'dead' when they quit the repository. Unfortunately, it is impossible to distinguish if the project has quitted the repository because it went independently, or moved to another repository, or just ended; thus, their disappearance does not necessarily relate to market demands, or product lifecycles, but to managerial premises. Finally, we look at those projects whose information on downloading activity is available. As the source of information comes from relational database, some observations could be missing in the joining process. We follow this procedure over twelve months running from February 2005 to January 2006, to build an unbalanced panel with 25,722 total observations.

3.2 Variables and analysis

Dependent Variable. Project success.

We present *project success* as a dependent variable; for that we use number of downloads per month as a measure of project success. A download means one user retrieves the executable files of the software for her private use; thus downloads are an output measure of success or popularity among a mass of users. Number of downloads is consistent with the nature of a software game as a digital good through Internet, because downloading is the sole way to access it; furthermore, it is a common measure of performance in OSS (Crowston *et al*, 2003; Grewal *et al*., 2006).

[4] GPL grants the programmers and users of software the privileges or freedom to distribute and modify copies of the software, and transfers those privileges to further developments.

Independent Variable. Social network as a source of social capital.

Social capital measures are largely heterogeneous within academic literature, thus they vary as they refer to different social capital perspectives [[5]].In this paper our measure of social capital lies in the structure of knowledge flows and refers to the sum of [complementary] resources attainable through a social network of projects that include both unidirectional and bidirectional relations. In our analysis, we test whether direction in knowledge flows in the social network of projects generate different outcomes in terms of the projects' success.

Here we measure directionality of ties among projects through identifying roles and affiliations, that is, contributors, members and duality roles. The presence of such ties indicates knowledge flows that enhance performance of the focal project.

We focus on dyadic relationships since they represent ties between pairs of projects linked through at least one of their members, or contributors. Nonetheless, if projects share more than one individual we count just one tie if they are the same typology of relationships. In doing so, we are measuring the relevance of structural diversity instead of the density of relationships. This practice also helps to diminish a plausible effect of excess downloading just for having more individuals involved in both projects simultaneously. For example, if the number of individual programmers from project ***A*** contributing to project ***B*** is three, it counts as one tie between projects ***A*** and ***B***. Thus, our measure for ties attempts to equate a linkage between the knowledge of the focal project and the knowledge of any other project within the SourceForge.net.

However, the important feature we want to test is the effect of knowledge flow direction on project success. We distinguish between those ties that receive *inbound* knowledge –or ties through individuals who play the 'contributor' or a 'duality' role in the focal project, from those ties, which provide *outbound* knowledge to other projects. All categories of ties measure knowledge flows from the focal project's point of view, therefore, although above we define nine forms of ties we can group those that represent the same directions of knowledge sharing for the focal project. In addition, we observe that a tie through individuals who simultaneously play as members and contributors in both projects does not actually represent a share of knowledge, but the common stock of knowledge for both projects, and thus we disregard this category. The independent variables are:

[5] For example, in Putnam's tradition, social capital measures membership in voluntary organizations as an indicator of networks together with norms and social trust (Putnam, 1995; Tsai and Goshal, 1998). In Granovetter's and Burt's tradition, social capital measures, to some extent, the information flows through exchanges among partners (Glaeser *et al*, 2000; Koka and Prescott 2002; Oh *et al*, 2004).

(1) *Active-inbound* ties: We consider the reception of knowledge from other projects toward focal project; nonetheless, other projects receive some share of knowledge. That means ties through non-members individuals that contribute to a focal project, and simultaneously contribute to other projects. It measures focal project ties through the forms 'contributor-contributor' and 'contributor-duality'.

(2) *Inactive-inbound* ties: We consider the reception of knowledge from other projects toward the focal project, but other projects do not receive any share of knowledge. It means ties through contributors to focal projects that are sole members in other projects. They measure focal project ties through the form 'contributor-member'.

(2) *Active-outbound* ties: We consider the provision of knowledge from the focal project toward other projects. It means ties through individuals members of the focal project that actively contribute to other projects. They measure focal project ties through the forms 'member-contributor' and 'member-dual' roles.

(4) *Inactive-outbound* ties: We consider potential knowledge flow for both focal and other projects, but none of them actually receives a share of knowledge. It means ties from individuals that are sole members in both focal and other projects. They measure focal project ties through the form 'member-member'.

(5) *Duality-inbound* ties: We consider the reception of knowledge from other projects toward the focal project, where individuals play a duality role and other projects do not receive any share of knowledge. It means ties through individuals that play a dual role as members and contributors in a focal project and are sole members in another. They indicate focal project ties through the form of 'duality-member' ties.

(6) *Duality-outbound* ties: We consider the provision of knowledge flow from the focal project, where individuals play a duality role while other projects receive some share of knowledge. It means ties through individuals that play a duality role as members and contributors in a focal project and contributors in another. They indicate focal project ties through the form of 'duality-contributor' ties.

Control variables

In the present study, we have included the project's characteristics such as control variables. On one hand, these control variables reflect the characteristics, which may induce differences in dependent variable because of demographic issues, such as size and age of a project; but also those referring to their technical peculiarities, which could demand an increase in knowledge transfer. Our control variables include:

(1) Project SIZE: Number of members of focal project.

(2) Project AGE: Age of the project in days.

(3) STAGE: OSS projects included in our database are at different stages of development, going from planning, pre-alpha, alpha, beta, production, mature and inactive. The status of the project and its ability to attract programmers, and therefore the number of ties and contributions from outside, are strongly associated. At its first stage, the core process is the creation of an initial system that will evolve over time. At the last stages, the core process is the diffusion of the product. Thus, we controlled for the evolution as the project may require different knowledge along its stages.

(4) Characteristics: We controlled for characteristics such as programming language, operating system, user interface, intended audience, and speaking language of the game.

(5) Previous project success: We control for the number of downloads at the previous period to take into account the possibility that successful projects attract the attention of more programmers. This is lagged dependent variable.

In order to test our hypotheses on a dynamic approach, we use a differences-in-differences fixed effects estimator; that is taking differences in both dependent and independent variables on a monthly basis. Therefore, we condition a variation in success at month $t+1$ with respect to month t, to a variation on ties at month $t+1$with respect to month t, controlling for the project's fixed effects over the period of study. We want to measure how a variation on a pattern of knowledge flows influences the monthly rate of downloading. The use of this type of regression method obeys our intention of testing how differential rates of knowledge exchange strategies condition project success. A variation in a particular type of tie means there is knowledge flow from or toward a different project, such as discovering a new source of knowledge, or a new channel for diffusion. The fixed effect estimator allows us to isolate the effect of a particular project, whose characteristics make it more attractive for programmers. The differences-in-differences fixed effects estimator allows us to isolate variations in dependent variable due to changes in independent variables conditional to the particular project's characteristics, which make them more attractive for programmers.

Restating our hypotheses in terms of our independent variables, we obtain a set of equations to test as expressed in Table 1.

	The effect on project success of:	is	than the effect of
H1.	*Active-Inbound* + *Inactive-Inbound*	>	*Active-Outbound* + *Inactive-Outbound*
H2.	*Active-Inbound*	>	*Inactive-Inbound*
H3.	*Duality-Outbound*	=	*Active-Outbound*
H4.	*Duality-Inbound*	>	*Duality-Outbound*

Table 1. Equations and expected results

4. Results

We have observed our data in a descriptive way, and furthermore we performed differences-in-differences fixed effects regressions to test whether our theoretical arguments hold. Table 2 reports basic descriptive statistics for the dependent and independent variables. The initial descriptive analysis indicates that projects, on average, report an increase of 20 downloads every month. Nonetheless, we observe that there is huge dispersion on this variable, as the minimum and maximum increase in downloads is approximately 400,000. Concerning the different ties in our study, we also observe an extremely low mean, but an important dispersion, except for the *Inactive-outbound* ties that do not present any variation at all, and finally they are not included in the regression analysis.

Variation in:	Mean	Std. Dev.	Min	Max.
Project success: Downloads	20.819	5004.585	-468651.500	463422.500
Active-inbound	0.005	0.095	-3.995	7.277
Inactive-inbound	0.010	0.209	-8.444	14.738
Active-outbound	0.003	0.063	-3.270	3.003
Inactive-outbound	0.000	0.000	0.000	0.000
Duality-inbound	0.002	0.159	-3.089	16.547
Duality-outbound	0.004	0.073	-1.178	5.458
Size	0.007	0.175	-8.618	10.279

Table 2. Basic descriptive statistics

Table 3 reports the results from differences-in-differences fixed effects on downloads. We addressed our empirical analysis by identifying knowledge flows among projects by studying diverse typologies of ties relating the possible 'member' and 'contribution' combinations. Here we show two basic regressions; Model 1 presents the basic model with controls. The control variables do not significantly explain variation in rates of downloads, except for the previous period difference in downloads, which has a negative effect on current period difference in downloads. This result is an indicator that projects present a decreasing trend in their success. Model 2 contains all our measures for ties as explanatory variables. We obtain that *active-inbound* ties have a positive and significant effect on the rate of downloads. It means there is a benefit from receiving assistance from individuals who play the 'contributor' role in focal project while they also contribute to others. We also find that *inactive-inbound* ties have a negative although non-significant effect. It implies there is no gain in just receiving knowledge from outside. The *active-outbound* ties show a surprising positive and significant effect; it means the focal project benefits from individuals who play the 'member' role while they are 'contributors' outside. The *inactive-outbound* ties drop from our regression, as this variable does not show variation over time. It means there is a fixed load of individuals who consistently do not contribute to any project. *Duality-inbound* ties have a negative and significant effect on project success; meanwhile, *duality-outbound* ties have a significant and strong negative effect on rate of downloads.

Dependent variable: *Project Success*	(1)		(2)	
Active-inbound			1517.380	***
			0.000	
Inactive-inbound			-97.452	
			0.455	
Active-outbound			1217.611	***
			0.001	
Inactive-outbound			-	
Duality-inbound			-337.702	**
			0.032	
Duality-outbound			-765.607	**
			0.031	
Project success $_{t-1}$	-0.490	***	-0.489	***
	0.000		0.000	
Size	-108.597		-225.611	
	0.502		0.167	
AGE	-36.543		-35.458	
	0.440		0.453	
STAGE	-159.904		247.215	
	0.935		0.899	
project characteristics controls included				
F	0.831		629.98	
P>F	0.000		0.000	
** significant at 5%	*** significant at 1%			

Table 3. Results from differences-in-differences fixed effects regressions

Viewed through our hypotheses, we can observe that this study confirms hypothesis 1, because the aggregate effects of *active* and *inactive inbound* ties (1419.48) are greater than the aggregate effects of *active* and *inactive outbound* ties (1217.611). Therefore, reception of knowledge is more favourable than provision for project success.

In hypothesis 2, we expect to have a greater effect from *active-inbound* ties than from *inactive-inbound* ties. Our result is consistent with hypothesis 2, and confirms that it is better for project success if individuals have unique insight during all the projects.

Hypothesis 3 establishes no [significant] difference among *duality-outbound* and *active-outbound* ties' effects on project success. Nonetheless, we find a stronger, positive effect of *active-outbound* ties compared to *duality-outbound* ties. Our results do not support hypothesis 3. Despite the fact that this result contradicts the theory and our expectations it may imply that those individuals serving as *active-outbound* ties attract the attention of other individuals because they open new channels for project diffusion or establish a good reputation on other projects.

We found that *duality-inbound* ties have a greater effect than *duality-outbound* ties, although both are negative; therefore, our results confirm hypothesis 4. It may mean that individuals that play the 'duality' role on the focal project while they contribute to other projects distract resources and attention from the focal project.

Overall results demonstrate that knowledge flows are significant for project success, and they build channels for flow on the extent that there is structural diversity.

5. Discussion and implications for future research

Social capital, generally defined as the actual and potential resources embedded in relationships among actors, is an important predictor of group and organisational performance (Adler and Kwon, 2002; Bourdieu, 1986; Leana and Van Buren, 1999; Nahapiet and Ghoshal, 1998). The structural dimension of social capital focuses on the nature and strength of relationships, and the communication flows embedded in networks of individuals and organisations. The advantages ascribed to social capital include better group communication, efficient collective action, enhanced stocks and use of intellectual capital, and better access to resources. Social capital also applies when answering questions about individual motivations; programmers consider it useful to maintain their social network alive and nurture the relation by providing their knowledge. This assertion is consistent with Nahapiet and Ghoshal (1998), who postulate that the relational dimension of social capital induces actors to formulate an expectation on the value of the resources they supply to the structure [6].

A network of OSS projects is fertile ground for testing and supporting the sources of social capital. Contributions to projects are the sole means for development and success, but their patterns differ widely across OSS projects; some projects attract a large number of contributors while others do not. There are projects in which most of the advances come from the voluntary contributions of their own members, while others rely on contributions from actors initially assigned to other projects. Ties between OSS projects represent the network structure, and

[6] Nahapiet and Ghoshal (1998) identify three dimensions of social capital that inhere in knowledge exchange and recombination: (1) structural, which reflects the impersonal pattern of ties; (2) relational, as a sense of trust, norms, obligations and expectations that actors develop along connections; and (3) cognitive, as the bonding force, such as shared understanding and identification that hold the group together. These three aspects of social capital combine to improve information transmission and absorption among organizational members, thus enhancing overall organizational performance.

therefore they are able to indicate differential levels of social capital based on individuals' affiliations and roles. We identified social capital in terms of the social network of the OSS as the number of ties generated through members and contributors and the role they play by solving programming gaps in both focal and other projects.

In this research, the straightforward outcome is empirical evidence concerning differences in social capital depending on the structural diversity of project members by identifying paths of communication across different projects. Moreover, empirical results also indicate that some social capital arguments do not hold as the effects of the role played by individuals is a significant source of variance in the access to network resources and project success. Explicitly, we find that project members playing a contributor' role in the focal project and in other projects do not contribute to success, but instead hamper it. We also find that knowledge and skills received through inbound ties lead to superior success, whereas providing knowledge and skills to other projects is beneficial to one's own success only if individuals play an active role as contributors in other projects beyond pure membership.

The core of our contribution relies on the measurement and identification of the collaborative structure – by identifying programmers' roles and affiliations - for knowledge sharing and the differential effects on project success related to knowledge reception and provision. As our own results show, this measurement supports the generalised reciprocity exchange theory; therefore, with the aim of finding significant sources of social capital, the necessity emerges for accounting differences in investment strategies or knowledge exchanges that complement the stock of capital.

However, one of the main limitations of the present study is that we do not measure the quality of the exchange, but only its direction. While we know inbound ties provide new knowledge and improved performance, we do not know anything about the quality of that knowledge. Thus, a project may need to tie to a broad collection of projects because of the poor quality of inbound contributions. Equally, in the case of outbound knowledge, we could assume the contributions of project members make them visible to the network, but we do not know how valuable their contribution is for others. Neither do we account for modularity of the project; a project with plenty of modules is more attractive to programmers because they clearly identify tasks and goals (Baldwin and Clark, 2006).

Another drawback involves our inability to link the different functions played by the contributors. We affirmed the presence of skill complementarities [that should be positively associated to project performance] in networks of cooperation across OSS projects; nevertheless, we do not measure complementarities but assume they occur through ties. Different participants perform different functions that facilitate the rapid change and creation of stable releases, including testing, contributing new changes, coordinating releases, and maintaining documentation. An important aspect of the collaborative approach is to help individuals find tasks in which they can better apply and exploit their talents. Moreover, self-selection and voluntary participation shape projects to reach the required level of complementary skills, which we were unable to include in our analysis.

We foresee a promising vein of research about social capital and OSS networks, as they present a symbiotic environment. Research on social capital will nurture findings derived from the study of open source software networks; meanwhile our understanding of the open source software phenomenon enriches us with a deep consciousness regarding the exchange processes for the creation of a valuable stock of social capital.

References

Adler, P.S., Kwon S., 2002, Social Capital: Prospects for a New Concept, *The Academy of Management Review*, 21(1):17–40.

Ahuja, G., 2000, Collaboration Networks, Structural Holes, and Innovation: A Longitudinal Study, A*dministrative Science Quarterly*, 45(3):425–455.

Allen, T., 1977, *Managing the Flow of Technology. Technology transfer and the Dissemination of Technological Information Within the R&D Organization,* Cambridge, MA, MIT Press.

Ancona, D.G. and Caldwell, D.F., 1992, Bridging the boundary: External activity and performance in organizational teams, *Administrative Science Quarterly* 37:634-665

Baldwin, C.Y., Clark, K.B., 2006, The Architecture of Participation: Does Code Architecture Mitigate Free Riding in the Open Source Development Model?, *Management Science*, 52(7):1116–1127.

Bergquist M, Ljungberg, J., 2001, The Power of gifts: organizing social relationships in open source communities, *Information Systems Journal*, 11 (4):305–320.

Bourdieu, P., 1983, "Forms of Capital", In Richardson, J.G. (Ed.), *Handbook for Theory and Research for the Sociology of Education*, New York, Greenwood Press**:** 241–258.

J.S. Brown and Duguid P., 2001, Organizational learning and communities-of-practice: toward a unified view of working, learning and innovation, *Organization Science* 21:40–57.

Burt, R.S., 1992, *Structural Holes: The Social Structure of Competition*. Harvard University Press, Cambridge, MA.

Burt, R.S.,2000, "The network structure of social capital", in B.M. Staw and R.I. Sutton: *Research on Organizational Behavior*, Amsterdam, London and New York, Elsevier Science JAI: 345–423.

Cohen, D. Zhou, 1991, Status Processes in Enduring Work Groups. *American Sociological Review* 56(2):179–188.

Coleman, J.S., 1988, Social Capital in the Creation of Human Capital, *American Journal of Sociology* 94: S95–S120.

Christley, S. and Madey, G, 2005,*"Collection of Activity Data for SourceForge Projects"*, Technical Report: TR-2005-15, Dept. of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN.

Crowston, K., Annabi, H, Howison, J., 2003, *"Defining Open Source Software Project Success"* 24th International Conference on Information Systems, http://opensource.mit.edu/papers/crowstonannabihowison.pdf

Cummings, J.S., 2004, Work groups, structural diversity, and knowledge sharing in a global organization, Management Science 50 (3): 352-364.

DiBona, C., Ockman, S., Stone, M., Eds., 1999, *Open Sources: Voices from the Open Source Revolution*. Sebastopol, CA, O'Reilly & Associates.

Ekeh, P.P, 1974, *Social Exchange theory: The Two Traditions*, Cambridge, MA: Harvard University Press.

Faust, D., 1997, Centrality in affiliation networks, *Social Networks* 19(2): 157-191

Franke, N. and Shah, S., 2003, How communities support innovative activities: an exploration of assistance and sharing among end users, *Research Policy* 32(1):157-178.

Glaeser, E.L., Laibson, D., Sacerdote, B, 2000, The economic approach to social capital, *National Bureau of Economic Research,* Working Paper 7728.

Granovetter, M., 1973, The Strength of Weak Ties. *The American Journal of Sociology*. 78(6):1360–1380.

Granovetter, M., 1985, Economic Action and Social Structure: The Problemm of Embeddedness, *The American Journal of Sociology,* 91(3):481–510.

Grewal, R., G.L. Lilien and G. Mallapragada, 2006, Location, Location, Location: How Network Embeddedness Affects Project Success in Open Source Systems, *Management Science*, 52 (7): 1043-1056

Gulati, R. and Gargiulo, M.,1999, Where do interorganizational networks come from? *The American Journal of Sociology* 104(5): 1439-1455.

Hansen, M., 1999, The search-transfer problem: Ther role of weak ties in sharing knowledge across organization subunits, *Administrative Science Quarterly*, 44(1):82–119.

Koka, B., Prescott, J., 2002, Strategic alliances as social capital: A multidimensional view, *Strategic Management Journal*, 23(9):795–816.

Kollock, P., 1994, The emercenge of exchange structures: An experimental study of uncertainty, comitment, and trust, *The American Journal of Sociology*, 100(2):313–346.

Krackhardt, D, 1992, 'The strength of strong ties: The importance of philos in organizations', in Lakhani K.R., von Hippel, E., 2003, How open source software works: "Free" user-to-user assistance, *Research Policy*, 32(6):923–943.

Lakhani, K.R., Wolf, B., 2005, "Why Hackers Do What They Do: Understanding Motivation and Effort in Free/ Open Source Software Projects", in J. Feller, B. Fitzgerald, S. Hissam, and K.R. Lakhani (eds.), *Perspectives on Free and Open Source Software*, Cambridge, MA: MIT Press: 3–21.

Larson, A., 1992, Network Dyads in Entrepreneurial Settings: A Study of the Governance of Exchange Relationships, *Administrative Science Quarterly*, 37 (1):76–104.

Lerner, J., Tirole, J., 2001, The open source movement: Key research questions, *European Economic Review*, 45(4):819–826.

Lerner, J., Tirole, J., 2002, Some simple economics of open source, *The Journal of Industrial Economics*, 50(2):197–234.

Leana, C., Van Buren, H.J.III., 1999, Organizational Social Capital and Employment Practices, *The Academy of Management Review*, 24(3):538–555

Levin, D.Z. and Cross, R., 2004, The strength of weak ties you can trust: the mediating role of trust in effective knowledge transfer, *Management Science* 50(11):1477-1490.

Lin, N., 1999, Building a Network Theory of Social Capital, *Connections*, 22(1):28–51.

Long, Y., 2006, Social structure, knowledge sharing, and project performance in open source software development. Dissertation project The University of Nebraska- Lincoln.

McKelvey, M., 2001, Internet Entrepreneurship: Linux and the Dynamics of Open Source Software, *Centre for Research on Innovation and Competition*, Discussion Paper No. 44, Manchester.

Nahapiet, J., Ghoshal, S., 1998, Social Capital, Intellectual Capital, and the Organizational Advantage, *Academy of Management Review* 23(2):242–266.

Oh, H.; Chung, M; Labianca, G., 2004, Group Social Capital and Group Effectiveness: The Role of Informal Socializing Ties. *Academy of Management Journal*, 47(6):860–875.

Pelled, LH, 1996, Demographic diversity, conflict and work group outcomes: An intervening process theory, *Organization Science*, 7(6):615–632.

Portes, A., 1998, Social capital: Its origins and applications in modern sociology. *Annual Review of Sociology*, 24 1–24.

Powell, W.W., 1990, Neither Market Nor Hierarchy: Network Forms of Organization. *Research in Organizational Behavior. 12:295–336.*

Putnam, R. D., 1995, "Bowling Alone: America's Declining Social Capital." *Journal of Democracy* 6(1): 65-78.

Raymond, E. S., 1999, *The Cathedral & The Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol, CA, O'Reilly & Associates.

Raymond, E. S., Young, B., 2001,. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol, CA, O'Reilly & Associates.

Sorenson, O. and Stuart, T. E., 2008, “Bringing the context back in: Settings and the Search for Syndicate Partners in Venture Capital Investing”, *Administrative Science Quarterly*, 53: forthcoming.

Takahshi, N., 2000, The emergence of generalized exchange, *The American Journal of Sociology*, 105 (4):1105–1135.

Tsai, W., Ghoshal, S., 1998, Social Capital and Value Creation: The Role of Intrafirm Networks, *The Academy of Management Journal,* 41(4):464-476.

Uzzi, B., 1996, The Sources and Consequences of Embeddedness for the Economic Performance of Organizations: The Network Effect, *American Sociological Review*, 61(4):674–699.

Uzzi, B.,1997, Social Structure and Competition in Interfirm Networks: The Paradox of Embeddedness, *Administrative Science Quarterly*, 42(1): 35–68.

Uzzi, B. and Lancaster, R.,2003, Relational embeddedness and learning: The case of bank loans managers and their clients, *Management Science* 49(4):383-399.

Waserman, S. and Faust, D.,1994, *Social Network Analysis: Methods and Applications*, Cambridge MA: Cambridge University Press

Wellman, B., Wortley, S., 1990, Different Strokes from Different Folks: Community Ties and Social Support, *The American Journal of Sociology*, 96(3):558–588.

Woolcock, M., 1998, Social Capital and Economic Development: Toward a theoretical synthesis and policy framework, *Theory and Society*, 27(2): 151–208.

Zaheer, A; Bell G., 2005, Benefiting from Network Position: Firm Capatiblities, Structural Holes, and Performance. *Strategic Management Journal*, 26(9):809–825.

Zajac, E., Westphal, J.D., 1996, Director Reputation, CEO-Board Power, and the Dynamics of Board Interlocks, *Administrative Science Quarterly.* 41(3):507–529.

Zhang, C, 2007, Emergence of New Project Teams from Open Source Software Developer Networks: Impact of Prior Collaboration Ties, www.mgmt.purdue.edu/academics/mis/workshop/cz_01907.pdf

To Contribute or Not To Contribute, An Exploratory Study on the Determinants for Contribution in OSS Projects.

Rebeca Méndez-Durón

Universidad Carlos III de Madrid, Departamento de Economía de la Empresa, c/ Madrid 126, Getafe, Madrid, Spain. 28903. E-mail: rebeca.mendez@uc3m.es

Keywords: Cooperation, Public Goods, Free-riding, Altruism, Collectives, Networks, Open Source Products, Software Industry.

JEL: H41, H42, D64, L14, L17, L86, P32

Introduction.

Open source software (OSS) mode of production is the object of study of many scholars, because it contradicts and questions many traditional concepts in economics and management theories, about motivations for contributing and hierarchical structures of governance. Since 2001, research in OSS addresses two branches. One devotes effort to demonstrate efficiency and productivity predominance of OSS over conventional software, while the other attempts to discuss the economic and management aspects that tune the innovation in such environment. Scholars stress the division by recognizing two mainstreams of analysis: better performance at lower cost and incentives under non-contractual forms of organization, (Dalle and David; 2004).

Disregarding the concern about superior quality of open source software, economics and management scholars struggle to find the drivers that lead individuals freely contribute to software generation, and the implications for innovation management. OSS production process conform a public good economy, where its development through a virtual network community exacerbates its public nature. Everyone benefits from contributing to software by developing code, individuals share also computer equipment, software repositories and operating standards, and every person who has access to the community rapidly obtains the outcome of others' provision (von Krogh and von Hippel, 2006). Goods are non-excludable, therefore, there is no loss if other individuals use it; on the contrary, goods gain value from greater user-base. As in any other public good economy, we can expect that some individuals may want to wait and see what other developers produce and then benefit from others' code for their own use, (Baldwin and Clark, 2006).

Despite a free-riding problem in OSS has never been discussed as an issue to avoid, there are many studies that, on the contrary, provide explanations about why people contribute. Scholars compare OSS to a gift-giving economy where individuals find ego-gratification in giving (Ljungberg, 2000), a generalized exchange economy where reciprocity and trust is the key driver for contribution (Hertel, 2003) or a scientific community where motivation is the personal gratification of finding a discovery or reputation acquired among their peers (Ljungberg 2000, Lerner and Tirole, 2001).

In a pragmatic view, earlier arguments claim individuals receive private benefits from contribution (Lerner and Tirole, 2002). These private benefits do come in the form of abstract but desirable goods, such as reputation and career concerns (Lerner and Tirole, 2001), self-fulfillment (von Krogh et al, 2003), or social approval (Ljungberg, 2000; Hertel, 2003). Other benefits are that individuals feel attracted to contribute in tasks where they can provide meaningful knowledge to mathematical and programming challenges given they have the appropriate skills to accomplish them (von Krogh et al., 2003) or because the atomized software architecture provide higher value from code production at lower levels of effort (Baldwin and Clark, 2006). Besides, the increasing number of firms that start engaging in OSS pay their programmers for developing code in open mode, therefore motives shift from intrinsic to extrinsic (Roberts et al., 2006). Most of these claims proceed from theoretical propositions (Baldwin and Clark, 2006; Lerner and Tirole, 2001); simulations (Dalle and David, 2003); case studies and longitudinal analysis of relevant single OSS projects (Bagozzi and Dholakia, 2006; Lakhani and von Hippel, 2003; Lerner and Tirole, 2002; von Krogh et al, 2003); and direct interviews and surveys about why contributors contribute (Hertel et al. 2003; Roberts et al, 2006).

Here we present an exploratory study of the determinants for contribution among individuals who decide to join a project for OSS development. We use a sub-sample of 3685 projects for the development of computer games over 12 months, from the Sourceforge.net repository, where 6939 individuals have signed as team members but not all of them contribute to software development. In our analysis, we have controlled for some motivators, such as wages and value derived from use, and included most of theoretical determinants for contributions, such as attractiveness of project, complementarities among contributions, and individual inclinations for contribute, such as preferences, search for reputation and role played by individuals. We use number of team members and number of outsider contributors for controlling free riding and find interesting effects: Individuals are less likely to contribute as the number of team members' increase, but are more likely to contribute as the number of outsider contributors' increase. These findings suggest the emergence of a governance-like structure that may exert control on the outcome of outsiders' contributions. The governance structure gains legitimacy as a meritocracy, and act as social control of production process (Eisenhardt, 1985), instead of applying a coercive authoritarian control that could deter other individuals' participation (Henderson and Lee, 1992; de Laat, 2007). Our findings open a new vein of research in OSS where governance and control go beyond a single project management authority and control mechanisms are example and behavior imitation.

Theories about cooperation and contribution in OSS.

OSS movement big question is still how programmers do contribute in code production. The production process of open source software requires the contributions of developers who donate their time, effort and knowledge to create pieces of code. Developers usually are volunteers or self-employed individuals and rarely there are contractual relations among them and the project to which they provide their contributions. Moreover, in open source networks, there are many individuals involved simultaneously in several projects, therefore, commitment between developers and projects rely on looser bonds where there is neither pecuniary reward in case of contribution nor punishment in case of defection (Roberts et al.; 2006).

Open source movement has the halo of a culture that encourages individuals to contribute and share their knowledge, with the promise that outstanding contributions will provide status and influence; nonetheless, there are few empirical explanations about why individuals contribute (Raymond, 1998). The general assumption is that OSS performs as a gift-giving economy where individuals gain recognition from others when providing gifts to other individuals in their community (Ljungberg, 2000). Although some defendants of OSS claim this as a good explanation (Raymond, 1998; Stallman, 1997), Ljungberg (2000) himself considers that similarities are too constrained, because in a gift-giving economy the gift usually is not a public good, and the recipient is the person who obtains benefits from the gift, while the gift-giver obtains benefits from the act of giving. The main characteristic of contributions to open source software is that they are gifts or provisions to public, non-excludable goods; therefore, the gift-giver provides the gift but also enjoys it. Ljungberg (2001) also compares OSS to a scientific community. He encounters drivers of OSS 'economy' relying on reputation, status, and career concerns, like job opportunities, and indirect pecuniary benefits; Lerner and Tirole, (2001) share these assumptions and confirm some of them at case studies from Apache and Sendmail projects. OSS contributions also represent a generalized exchange economy supported by the expectation that if a community member provides assistance today, someone else will provide him with assistance when he needs it (Mendez-Duron and Garcia, 2008).

OSS projects' collaborative structure is based on cooperation by knowledge sharing defined here as the provision or reception of information, expertise, and feedback regarding the development and/or modification of a source code.

According to scholars who study cooperation from a sociological perspective, cooperation involves three conditions: it is costly to the contributor, it is valuable for others who receive the contribution, and it is voluntary (Holländer, 1990). In gift giving economies, the others receiving the gift manifest sympathy and gratitude towards the contributor, and these actions generate a sentiment of social approval for the contributor (Holländer, 1990). An individual that is part of a group will exert an effort to contribute to public good provision if other members in the group also exert a similar effort (Sugden, 1984). The realization of an effort will be dependent on the reciprocity and exchange within members of the group; individuals feel an obligation towards other members if they benefit from public good. Kollock, (1993) posits that relaxed systems to measure the extent of contributions lead to more stable exchange relationships. Relaxed systems rely on trust[7], assign differential threshold levels at which individuals are supposed to contribute and assume that contributions can be of diverse nature; therefore, an individual does not need to receive the exact amount and kind of contribution that she has given away. Even under these conditions, it would be naïve to think that individuals in OSS act disinterestedly or just for social approval, because an individual, by giving code create the obligation on recipients to give other code back, although repayment is not direct nor immediate (Zeytlin, 2003).

From an economic perspective, an individual would contribute to a public good if contribution were the best response to others' strategies. An individual will contribute onto a public good if her expected value is higher than her reservation utility; if she expects to have superior rewards from contribution, rather than devoting effort to something else; or if risk associated to contribution is lower than the level of expected utility obtained from public good, (Harstad, 1980).

Lerner and Tirole (2001) shed light onto economic conjectures about contribution in OSS. They claim that OSS do not necessarily contradict economic paradigm, because individuals

[7] Kollock denotes trust as credit as in the case of accounting, where an individual would be indebted towards others by creating the bonding of trust.

can gain a private benefit from contribution and their reservation value is greater than their effort. They reinforce the idea that many of motives that lead programmers to contribute refer to reputation and career concerns.

From a psychological perspective, in OSS, individuals may have multiple motivations for contribution and these motivations may range from intrinsic to extrinsic (Roberts, et al, 2006). Among the many explanations for contribution, intrinsic motivations are those that satisfy basic human needs, like contributing for personal curiosity to solve mathematical puzzles, or being part of a community (Lakhani et al, 2003). Extrinsic motivations are those applied or imposed by an external source that rules the impulse to contribute, like gaining reputation, improving a career or being paid (Roberts et al. 2006).

When a collectivity is supposed to contribute, size of the collectivity always matters. Even in a group of size two, one of the individuals would have incentives to free ride and under-provide public good, (Smith, 1980; Isaac and Walker, 1988; Offerman et al., 1996). One of the main arguments is that when an individual perceives the effectiveness of her contribution decreases, her willingness to contribute also decreases. Such perception increases when the size of collectivity increases. In open source software provision, there exists a serious temptation to free ride on others' contributions. Programmers can wait and see how other people contribute and later benefit from these contributions.

Solutions to reduce free riding effect on teams are separating ownership from labor (Holmstrom, 1982; Maitland et al, 1985); or creating competition among team members (Holmstrom, 1982). The first solution, although quite common in private firms, it is very unlikely to apply to open source software, as by definition, ownership belongs to nobody. The second solution is more likely to apply in our setting, if individual contribution is to be compared with others performance, and the reward or punishment come from differential access to private benefits. If the contribution terms seem equitable for all individuals then, contribution will occur (Maitland et al. 1985).

Kollock (1998) provides three types of solutions for contribution, motivational, strategic and structural solutions. Motivational solutions imply actors are not totally egotist, while strategic solutions deal with private benefits from contribution and cooperation. Finally, structural solutions imply a change in structure that leads to higher or lower contribution rates.

Drivers for contribution.

Different theories have given various arguments that support motivations for contribute. All these views seem to be complementary, because evidence shows that there is no single driver for contribution; on the contrary, individuals have several simultaneous motivations to contribute, (Hertel et al., 2003; Roberts et al., 2006; Zeytlin, 2003). Several authors hypothesize about the drivers for contribution in OSS; here we summarize some of their arguments.

Use-value of the project. Since their start, projects usually have an objective, goal or milestone to accomplish, and somewhat they follow a plan. It is plausible that, depending on project ambitions, the planning is tight or loose in terms of tasks. Overtime, projects will accomplish or not their goals, but in all cases, individuals can infer from their planning if projects attire their attention because they can obtain a value from its use or because they pose interesting programming puzzles which challenge their abilities (Hertel et al., 2003; Lerner and Tirole, 2002; Raymond 1999, Roberts et al., 2006). The more use- value individuals obtain from project, more likely are to participate.

Search for personal reputation and career concerns. The most remarked argument from economic scholars about incentives to contribute is the search of personal recognition. Lerner and Tirole (2002) study deals with reputation, career concerns and ego gratification as theories to explain individual propensity to contribute. They studied the enrolment of professionals in Apache, Perl and Sendmail projects and found that signalling incentives play a major role as individual motivation. Hertel et al. (2003) study individual incentives of Linux developers to participate in the kernel project -main architecture platform of Linux-; in psychological perspective, they found that main drivers for contribution are individual reputation and social comparison. Also von Hippel and von Krogh (2003) and von Krogh et al. (2003) finds that newcomers need to adapt when they are about contributing in official pre-defined modular software architecture, but they are more likely to contribute when the architecture design is still somewhat open. Nonetheless, if individuals are trying to signal themselves as good programmers, they would like to contribute soon in the project, to obtain sooner the benefits from reputation. Therefore, individuals who seek a strong reputation as programmers are more likely to contribute.

Individual creativity. Individuals who need that some software characteristics are quickly developed are more likely to be innovative. Von Hippel, (2003) denominates these individuals as lead users and postulate they are great sources for innovation. Lead users are more likely to be innovative because they experience needs for products much before the majority of other users. Roberts et al. (2006) also test for intrinsic motivators to contribute, considering individual risk preferences and creativity as such motivators. Creativity helps to solve problems, and in particular, to design appropriate pieces of code. Moreover, the open nature of the innovation process in OSS promotes that individuals bring their innovations towards projects because the absence of routines do not hamper that contributors rapidly observe the outcome of their innovation (Nelson and Winter, 1958). Therefore as individual creativity increases, the likelihood of contribution increases.

Individual Skills. Individuals who have specific intellectual capital to develop and solve specific tasks are more likely to participate (Maitland et al., 1985) due to costs of contribution have diminishing returns to intellectual capital investments. Individuals may participate in software as they have the skills and expertise demanded by the project. If individuals feel they are too far from the level of expertise required, they hardly could make significant contributions, and then their willingness to participate would erode. Giuri et al., (2004), find also that there is some self-selection division of labour within OSS projects, and that individuals choose to contribute in those activities who match their own skills. Therefore, individuals that are more skilled are more likely to contribute.

Complementarities among contributions. Relying on the same principles of labour division as before, one may think that contributions are somewhat complementary. Baldwin and Clark (2006) proposed that projects with high modularity and in their study, high option value are more likely to attract a larger number of individual programmers, and that they actually contribute. Their argument underlies that in the absence of overlap in software creation, if the value of code produced is higher than the cost or effort to produce it, willingness to contribute will increase. Value of code produced increases with modularity, therefore when many other contributors provide their knowledge in small, well-defined tasks, that ensemble together into a modular architecture, the value of total code will increase. When relaxing the non-overlap in software creation assumption, the incentives for contribution decrease, because many other programmers could provide an alternative appropriate solution to the same problem. It means that if an individual perceives that her contributions are complementary to others', individuals are more likely to participate, whereas contributions are substitute, individuals are less likely to participate.

In addition to the factors above mentioned, there could be other that may structurally affect contribution to OSS, rather than modify the willingness to contribute. We consider these factors as accessory and not determinant in our model, but should not be disregarded.

Time availability. If individuals face some time constraints, it is very likely they would reduce their production of code. It would imply that if individuals face compromise to their individual work, personal circumstances and an active profile of contribution, their likelihood to contribute would decrease.

Intellectual property protection schemes. Trust in other individuals' behaviour about appropriation of software would drive the willingness to contribute, as contributors may fear that other users or commercial companies might exploit own efforts. The emergence of licenses that restraint for appropriation, -although they allow for free copy, modification and distribution, would lead to a higher willingness to contribute. Therefore, as more restrictive is the project license in terms of appropriation, more likely individuals will contribute.

Pay. Recently, an increasing number of firms have engaged in OSS creation. Programmers of theses companies receive a performance pay as in any other type of industry, but as Roberts et al. (2006) find, pay substitute intrinsic motivations for contribute. Nonetheless, we can expect that monetarily rewarded individuals are more likely to contribute.

Data and Methods.

The analysis of contribution in OSS demands a thorough database that from one side provides intricate details of project formation, organization and operation, and from other side details about the implication of individuals in the software development. Moreover, data should provide some evidence - or at least to allow for control, of the individual drivers for contribution.

3.1. Data setting

The data we use in this analysis come from the SourceForge.net Research Data, gathered by Department of Computer Science and Engineering, University of Notre Dame. SourceForge.net is the largest repository of open source software; it hosts over 170,000 projects and gathers over 1.5 million of registered users. SourceForge.net belongs to OSTG, Inc, who has shared activity data with the University of Notre Dame for the purpose of academic research on OSS, under the condition to share further the data to other researchers interested in open source software phenomenon (Christley and Madey, 2005).

Sourceforge.net as a repository has several characteristics that promote network exchanges. The purpose of repository is to provide a platform for software development over a worldwide web infrastructure for knowledge exchange. They host projects and provide tools that allow asynchronous communication, stock people's contribution, and screen software from redundant knowledge. To host a project in Sourceforge.net, an initiator should register on the network; and post a message to the platform indicating the type of software, its purpose and target public, the intellectual property regime, the programming language, the user interface, the phase of development and the team workforce who will be responsible for developing and controlling the software. Then Sourceforge.net administrators authorize the creation of a space for hosting the project, if it satisfies the main premises for the repository. Once the repository hosts the project, then it registers and controls all its activity. Every time there is a movement on the project, the repository electronically archives the information; these movements may include communications among members, forum posts, and more important, the artifacts –or modules of software code, produced by the people engaged in programming activities. As an important feature of a repository is to open the access to projects to everybody on the worldwide web, every single person, registered or not on Sourceforge.net, may see the project, look at the code, and contribute. The repository records changes in the team workforce, so it is possible to know if there are new members or if some of them have already quitted the project. The repository also records all the contributions of people outside of members of the project. As repository tracks all artifacts records, we can identify who are who are members of the project and who of them actually contribute.

Particularly, our dataset is a sub-sample of 69693 valid observations over twelve months of projects aimed at developing games' software. To get this sample, we look at the monthly dumps of data and select all projects that belong to the category of GNU Public License (GPL)[8]. We restrict the sample to such license to guarantee that projects were not subject to any restriction for copy, adoption and distribution, so all software is equally prone to reach the same audience. Constraining our sample to this license allows us to control for differences in intellectual property protection regime; therefore, we will test the likelihood to contribute in projects with comparable intellectual property schemes. Then, we limit the sample to games in a general category. This selection obeys to the purpose of tapping an appropriate control for use-value. Individuals that contribute into a software game are more likely to search amusement rather than solely solve a specific programming puzzle. Further, we restrict the sample for the projects that were alive during the whole sample, we make this decision because dataset classifies a project as 'alive' –when the repository still host the project, and 'dead' when they quit the repository. Unfortunately, it is impossibly to distinguish if the project has quitted the repository because it went independently, or moved to other repository, or just ended; thus, their disappearance does not necessarily relate to market demands, or product lifecycles, but to managerial premises. In addition, we look at those projects whose information on downloading activity is available, as the source of information comes from relational database, some observations could be missing in the joining process; we want to control for possible crowding effects due to successful projects that can attract a higher number of programmers. Finally, we should remark that none of these projects pay their developers for contribute; therefore contribution is altruistic in the sense of pure monetary rewards. We follow this procedure over twelve months running from February 2005 to January 2006, to build an unbalanced panel with 69693 total observations. With the use of these data, we can control for many of the variables used by other researchers, and we can scrutinize deeper less studied variables and search for other possible explanations.

3.2. Variables and analysis

[8] GPL grants the programmers and users of software the privileges on freedom to distribute and modify copies of the software, and transfers those privileges to further developments.

Dependent Variable. Individual contribution of member i in project J. (Contribute)

We look for likelihood for contribution in software projects for those individuals who has signed as members of the project. An individual acquires membership condition when she asks the project manager to belong to the core team and thus, her name appears in the front cover of the project. In addition, we want to control for project characteristics that make it prone to contribution. For every single software project in our sample, we have recorded when each individual *i* that is member of the project *J* has sent code. We have grouped contributions in months, therefore we have identified whether an individual has contributed to a particular project *J* in month *t*. For every single month, we record individual contribution as a dichotomous variable, with value 1 if individual *i* has contributed to project *J* in time *t*, and value 0 if individual *i* has not contributed to project *J* in time *t*.

Independent variables. Drivers for contribution.

Search for personal reputation and career concerns (first project). Personal reputation comes from the signal the individual provides about her quality. In our dataset we have information about individuals within the network, we cannot observe the outside-the network world. Therefore, we cannot directly relate the total effect of contribution to individual reputation. Nonetheless, we can observe if the project of study is the first project, in which the individual appears as member. We use an indicator variable of value 1 if this project is the first for the individual, and zero otherwise. If individuals concern about reputation and invest in a signal, participating in their first project as soon as they enter would be a reliable measure of the signal. If the contribution is poor, everyone within and outside the network would realize about it; therefore contribution in the first project is costly, and only those skilled programmers can afford to send the signal. Contribution in first project satisfies all the conditions for effective signalling, it is costly, but those with higher programming skills face a lower cost.

Individual creativity (code contributions). We consider two different measures of individual creativity: First, the total number of artifacts, or packages of code, a member sends to any project at time t. Second, the aggregate overtime total number of artifacts, a member has sent to any project at time t. These measures accounts for how the individual is likely to propose tasks, as well as detect and solve bugs in a given period. The first measure, only looks at the pieces of code sent without taking into account how many contributions has make up to that date. The second measure takes into account all contributions made by individual.

Complementarities among contributions (COMPLEMENTARITIES). Each piece of code, or artifacts, is classified in categories. These categories could be general or specific to the project. We count the total number of categories for a project, and we construct a ratio of the categories in which individual *i* does not contribute divided by the total number of categories. If this ratio increases, it means that other individuals care for more categories in the project, while if the ratio decreases, it means that individual *i* care for the most of categories.

Individual Skills (SKILLS). We use also categories of artifacts. We count the number of categories in which each individual *i* contributes in all projects within the network and divide by the number of projects in which individual *i* contributes. If the ratio is greater than one, individual is able to contribute into more than one category per project, therefore is more skilled than other individuals that solely contribute to one category or less.

Control variables.

Time availability (code contributions squared). Our database does not allow us to infer other activities beyond the network on which the individual could have compromise. Nonetheless, we can observe the level of contribution within the network. We measure time constraints as the square of the number of artifacts that an individual sends to all projects in that period.

Number of members of the project (members). We want to control for the propensity to free ride a member of the project could have by relying on the contributions of other individuals who sign as members of the project.

Number of outsider contributors of the project (other contributors). We control for the propensity to free ride a member of the project could have by relying on the contributions of other individuals who are not members of the project, but are willing to contribute.

Role in the project (manager and developer). We control for the member that the project manager assigned to each member. We include the most common categories, project manager and developer, and use as base of comparison any other role. We use the dichotomous variable manager with value 1 if the individual plays the project manager role, and zero otherwise. We use the dichotomous variable developer with value 1 if the individual plays the project developer role.

Project performance (downloads). We consider the aggregate past performance of the project in terms of its number of downloads as control for the possible attractiveness a project may have for its own performance.

We use a fixed effects panel data estimator for the dichotomous variable Contribute. We use the logistic regression that allows us to control for fixed effects of the pair individual i - project J over the twelve-month period. We also run a logistic regression as if data were cross-section just for control.

Results.

Figure 1 shows the distribution of individuals among projects. In the graph, we observe a skewed distribution of individuals; in particular, 2356 projects have only one member, meanwhile there are approximately 50 projects that have more than 10 members. Concerning individuals, 6701 are members solely in one project, while the rest are members in no more than four projects simultaneously. In table 1 we present descriptive statistics of main variables, whereas in table 2, we present their correlation matrix. Table 3 shows the robust regression results, model 1 and 2 are logistic cross-sectional data over all the observations. We notice that COMPLEMENTARITIES has positive sign and is significant at 5%, it confirms the main arguments regarding individuals are more likely to contribute if there are some complementarities among contributions. Individuals observe in how many categories other individuals participate, and if this measure increases, she is more likely to contribute, therefore individuals foresee that there would be complementarities with her own contributions.

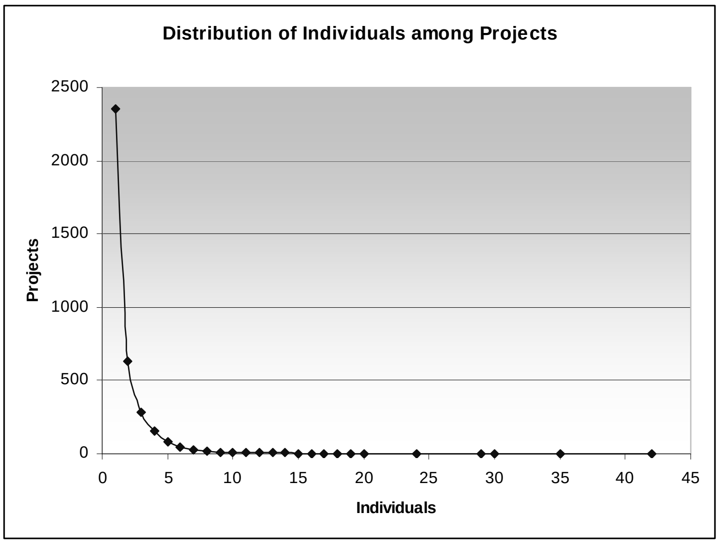

Figure 1. Distribution of individuals among projects.

Creativity, measured by code contributions would influences positively the likelihood to contribution, because individuals have personal characteristics that increase their willingness to give away ideas and solutions for problem solving, unfortunately this factor is not significant. Total Artifacts Squared represents time constraints of individuals with respect to active contribution within the network. Its coefficient resulted negative and but non-significant, even, the result indicates that contribution has a U-shape on Total Artifacts. If the individual faces his first project, the likelihood to contribute is very high, and has a strong significant effect. This reveals the desire to quickly signal in the network. Regarding the control variables, we find the negative and significant effect in number of members, but significant and positive effects in number of other contributors. It seams that members of a project would free ride in own colleagues but feel reinforced to contribute when outsiders also contribute. Being a developer and a manager is also significant to contribution, although, as expected, the effect of being a developer is higher than being a manager. Overall, past project performance has no effect on individual contribution. We included the squared term of total artifacts in order to control to decreasing returns to scale. We find a significant negative effect of total artifacts squared, implying that individuals would be less likely to contribute if as the number of total artifacts increase.

Models 1 and 2 presented a pseudo-R2 of 0.69, it means that these variables explain almost 70 percent of likelihood of contribution.

Models 3, 4 and 5 are the fixed-effects logistic regression results, studied over a period of 12 months. Results confirm the cross-sectional analysis, except for the roles of developer and manager, that counting overtime effects, there is no significant likelihood to contribute.

	Mean	Std. Dev	Min	Max	N
Contribute	0.010	0.101	0	1	69643
Total artifacts	1.809	52.268	0	3276	69643
Total artifacts ^{2}	2735.217	142481.300	0	1.07E+07	69643
Code contributions	0.221	3.114	0	225	69643
Code contributions ^{2}	9.745	425.253	0	50625	69643
First project	0.021	0.143	0	1	69643
Members	4.545	5.994	1	42	69643
Other contributors	0.085	3.861	0	357	69643
Developer	0.274	0.446	0	1	69643
Manager	0.107	0.310	0	1	69643
Downloads	9063.099	82549.130	0	2109955	69643

Table 1. Descriptive statistics.

	1	2	3	4	5	6	7	8	9	10	11
1 *Contribute*	1										
2 *Total artifacts*	0.3093 *	1									
3 *Total artifacts* ^{2}	0.1737 *	0.9354 *	1								
4 *Code contributions*	0.2972 *	0.2343 *	0.1474 *	1							
5 *Code contributions* ^{2}	0.1384 *	0.1424 *	0.0918 *	0.8552 *	1						
6 *First project*	0.3989 *	0.1461 *	0.094 *	0.338 *	0.1295 *	1					
7 *Members*	-0.0253 *	0.0093 *	0.0086 *	-0.004	0.0012	-0.026 *	1				
8 *Other contributors*	0.2023 *	0.4956 *	0.4916 *	0.1369 *	0.0862 *	0.0762 *	0.0018	1			
9 *Developer*	0.0255 *	0.0092 *	0.0048	0.0015	0.0027	-0.0109 *	0.1485 *	0.004	1		
10 *Manager*	0.0065	-0.0034	-0.0028	-0.0012	-0.0039	0.0014	-0.0583 *	-0.0002	-0.2129 *	1	
11 *Downloads*	0.0297 *	0.0201 *	0.0048	0.0155 *	0.0055	0.0039	0.0627 *	0.0103 *	0.0088 *	-0.0161 *	1

Table 2. Correlation matrix.

Table 3. Cross-section Logistic and Panel Data Fixed Effects Logistic Regression Results.

Dependent variable= Contribute $_t$

	Fixed effects logistic regression estimates panel data analysis		
	1	2	3
FIRST PROJECT		4.0458 ***	3.8495 ***
		(0.5100)	(0.4988)
CODE CONT		0.0003	
		(0.0492)	
CODE CONT CUMULATIVE			-0.0460 **
			(0.0198)
COMPLEMENTARITIES		1.6670 **	1.6697 **
		(0.7509)	(0.7887)
SKILLS		3.9554 ***	3.9949 ***
		(0.9475)	(0.9739)
CODE CONT ^{2}		0.0000	-0.0001
		(0.0005)	(0.0001)
MEMBERS	-3.4099 ***	-1.9413 ***	-2.0830 ***
	(0.2672)	(0.2491)	(0.2699)
OTHER CONTRIBUTORS	2.6395 ***	0.4877 ***	0.5647 ***
	(0.1608)	(0.1821)	(0.1941)
DEVELOPER	-3.0464 *	1.1297	1.3969
	(1.7567)	(2.9821)	(3.0109)
MANAGER	-1.6938	0.6196	0.6385
	(1.3662)	(2.4698)	(2.5282)
DOWNLOADS	0.0001 **	0.0001	0.0001 **
	(0.0000)	(0.0001)	(0.0000)
TOTAL ARTIFACTS		0.0142 **	0.0150 ***
		(0.0061)	(0.0053)
TOTAL ARTIFACTS ^{2}		-3.73E-06	-3.07E-06
		(0.0000)	(0.0000)
N	5841	5841	5841
n	509	509	509
degrees of freedom	5	12	12
Likelihood Ratio χ^2	2174.00	2362.35	2355.35
p> χ^2	0.0000	0.0000	0.0000

*Standard errors in parenthesis, * sig. 10%, **sig 5%, ***sig 1%*

Conclusions.

It is difficult to as even in the case where individuals have a strong personal drive for contribution, they would choose a project that is at least attractive for own use, and where their contributions receive recognition and no one else can appropriate their effort. Here we list and discuss some of the drivers for contribution and complement with the theories and previous results.

Contrary to previous studies, we do not find a significant effect of being newcomer in the probability of contribution. This finding could be related to complementarities on knowledge production. A newcomer does not contribute early in her the project until she is sure her contribution could really bring new knowledge to the project.

Smith (1980) considers the idea that when in the presence of mechanisms for exclusion from the benefits of public goods, then it is possible to induce contribution in a decentralized environment. The idea is compatible with our setting, if members of the project consider that in the presence of outsider contributors they will get lower payoffs because they will be excluded of the benefits of reputation and self-fulfillment then they are more likely to contribute.

Control relationship between project manager and team members is one of the central factors to effective performance in information systems teams (Henderson and Lee, 1992).

Control is the process of creating and monitoring rules through hierarchical authority; it restrict employees activities by regulating patterns of interaction and feedback mechanisms. Control is diffused among members. Control structures and mechanism derive mainly from task predictability (March and Simon, 1958); as task predictability moves from routines to uncertain behaviors, control moves from hierarchy to decentralized team-member control. Highly descentralized teams with open communication channels often fail to finish their tasks on time. Managerial behavior control refers to the extent tat the manager monitors and evaluates team members' behavior in order to assist them. Managerial outcome control is the degree to with the manager monitors and evaluates only the outcome produced by the team members. Team members exhibit both self control and outcome control. An individual exercises freedom or autonomy to determine both what actions are required and how to execute these activities. Team members exercise control over how they accomplish their task. Self-control is likely to be implemented when the organization cannot adequately measure behavioral performance or standardize transformation procedures. Performance of technically oriented development teams can improve if team members engage in increased levels of self-control. Control mechanisms that guide the team towards achievement of organizational goals are employed by a team member rather than a manager. Individual team member might choose to reallocate his or her efforts among tasks or even employ alternative work procedures or methods without advice or feedback from project manager. Team-member outcome control is defined as an attempt to influence the performance of the team by providing feedback on performance goal-related outcomes (Henderson and Lee, 1992).

An organization can tolerate a work force with highly diverse goals if a precise evaluation system exists. In contrast, a lack of precision in performance evaluation can be tolerated when goal incompatibility is minor. The choice between the tow is driven by the ease of performance evaluation

They recognize the chaotic organization of projects does not exclude the presence of leadership, but usually, this kind of control relies on meritocracy. Lerner and Tirole opened the gate to research in open source software, attracting academic attention to governance mechanisms, legal aspects and individual reputation and career concerns.

Nonetheless, this study has several limitations. We cannot observe specialization. The only specialization I observe is to become project manager or software developer; in any of these cases, these roles are significant. We cannot observe modularity, just the amount of artifacts a project receives, then this is an imperfect measure of modularity. If the code architecture were transparent then, I could discriminate among those projects who are more modular or not, and their actual effects. The measure of total artifacts tells two stories in opposite directions. First, it could be because the project is actually more modular, or it could be the project has poor quality. Failing to discriminate these two settings leads to non-significant results.

In this study we have considered that contribution is just a single choice, and that drivers for contribution depend on individual motivations; actually contribution implies drivers in two steps: incentives to innovate plus incentives to reveal innovation. Nevertheless, it would be important to disentangle these steps, to understand which for innovators the drivers to reveal information are.

Bibliography.

Bagozzi, Richard P; Utpal M Dholakia. 2006. Open Source Software User Communities: A Study of Participation in Linux User Groups. Management Science 52, no. 7 (July 1): 1099-1115.

Eisenhardt, Kathleen M. 1985. Control: Organizational and Economic Approaches. Management Science 31, no. 2 (February 1): 134.

Harstad, Ronald M., Marrese, Michael. 1982. Behavioral Explanations of Efficient Public Good Allocations. Journal of Public Economics 19, no. 3 (December 1): 367.

Henderson, John C., Lee, Soonchul. 1992. Managing I/S Design Teams: A Control Theories Perspective. Management Science 38, no. 6 (June 1): 757.

Hollander, Heinz 1990. A Social Exchange Approach to Voluntary Cooperation. The American Economic Review 80, no. 5 (December 1): 1157.

Holmstrom, Bengt. Moral Hazard in Teams. (1982) The Bell Journal of Economics, Vol. 13, No. 2 (Autumn, 1982), pp. 324-340

Isaac, R. Mark, Walker, James M. (1988). Group Size Effects in Public Goods Provision: The Voluntary Contributions Mechanism. The Quarterly Journal of Economics 103, no. 1 (February 1): 179.

Kollock, Peter (1993). "An eye for an eye leaves everyone blind": Cooperation and accounting systems. American Sociological Review, 58(6), 768.

Kollock, Peter (1998). Social dilemmas: The anatomy of cooperation. Annual Review of Sociology, 24, 183-214.

Paul B de Laat 2007. Governance of open source software: state of the art. Journal of Management & Governance 11, no. 2 (April 1): 165-177

Maguire, Stephen 1999. The discourse of control. Journal of Business Ethics 19, no. 1 (March 1): 109-114.

Maitland, Ian, Bryson, John, Van de Ven, Andrew H..(1985). Sociologists, Economists, and Opportunism. Academy of Management. The Academy of Management Review, 10(1), 59.

Offerman, Theo, Sonnemans, Joep,Schram, Arthur. (1996). Value orientations, expectations and voluntary contributions in public goods. The Economic Journal 106, no. 437 (July 1): 817.

OUCHI, WILLIAM C. 1977. THE RELATIONSHIP BETWEEN ORGANIZATIONAL STRUCTURE AND ORGANIZATIONAL CONTROL. Administrative Science Quarterly 22, no. 1 (March 1): 95.

OUCHI, WILLIAM G., DOWLING, JOHN B.. 1974. DEFINING THE SPAN OF CONTROL. Administrative Science Quarterly 19, no. 3 (September 1): 357.

Jeffrey A Roberts, Il-Horn Hann, Sandra A Slaughter. 2006. Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects. Management Science 52, no. 7 (July 1): 984-999.

Smith, Vernon L. (1980). Experiments with a Decentralized Mechanism for Public Good Decisions. The American Economic Review, 70(4), 584.

Sugden, Robert. (1984). Reciprocity: The Supply of Public Goods Through Voluntary Contributions. The Economic Journal 94, no. 376 (December 1): 772

von Hippel, Eric. (2007). Horizontal innovation networks--by and for users. *Industrial and Corporate Change*, 16(2), 293-315.

von Krogh, Georg, Eric von Hippel. 2006. The Promise of Research on Open Source Software. *Management Science* 52, no. 7 (July 1): 975-983.

Contributing by Doing Nothing… at all?

Rebeca Méndez-Durón,

Universidad Carlos III de Madrid, Departamento de Economía de la Empresa, c/Madrid 126, Getafe, Madrid, Spain, 28903. E-mail: rebeca.mendez@uc3m.es

Organizations rely in developing and sustaining a reputation to charge higher prices or to increase demand. Understanding and creating the processes and mechanisms of organization's reputation is in the interest of academics and practitioners. In this study, we study separately social interaction and the creation of a pool of high quality productive assets as determinants for organizations reputation. We also find that changes in allocation of productive assets are possible deterrents for having a good reputation, but social interaction and assets quality moderate their effect. Organizations that distract productive assets for other purposes have lower success; nonetheless, if they have created a good network of interaction and have a good pool of productive assets, the negative impact on reputation will decrease.

Keywords: Reputation, Asymmetric and Private Information, Networks, Open Source Software, Software Industry, Organization of production.

JEL: L14, L17, L23, L86, D23, D82

I. Introduction.

Reputation is an intangible asset on which organizations rely for acquiring a better position in the markets they compete. Its intangible nature makes it hard to measure and to assess its effect on organizations success. Nonetheless, when hearing about one organization's reputation, we can make some assumptions about their products or services quality. Reputation reveals to some extent information about organizations helping to distinguish their position relative to other players within the market where they play. Consequently, high reputation would lead to better opportunities for organizations, whether in the form of higher earnings, improved access to market for inputs, and higher market share in produce markets (Fombrun and Shanley, 1990; Podolny, 1993).

An important feature of reputation is that it represents two dimensions of perceived quality: First, it informs how consistently an organization performs; second, it informs about how other actors perceive its performance (Fombrun and Shanley, 1990; Kilduff and Krackhardt, 1994; Podolny, 1993; Rindova et al. 2005). The first dimension is an inference from perception about quality by directly observing organization's past performance; meanwhile the second is an inference of the organization's quality from referrals or endorsements by others. From direct observation, reputation involves a signal that allows separating high from low product quality organizations (Spence, 1973). The signal comes from the perception about organization's quality based on previous performance (Kilduff and Krackhardt, 1994). Moreover, in order to maintain a reputation, organizations have incentives to invest in a reliable and effective signal that requires to be sustained over time; therefore, actors care about self-reinforcing their reputation by continuously developing high quality activities (Fombrun and Shanley, 1990; Fudenberg and Levine, 1992; Kim, 1996; Tirole, 1996). From indirect observation, reputation [[9]] comes in the form of an endorsement, that reveals organization's social position on a hierarchical structure that orders players within an industry according to their qualities (Gould 2002; Podolny, 1993; Rao, 1994; Stuart et al., 1999). The endorsement emerges from interaction among organizations and other actors within a hierarchy, market, industry or network. Structures allow inferring information about organizations' quality from social relations (Fombrun and Shanley, 1990; Kilduff and Krackhardt, 1994). Therefore, hierarchy is a product of social process that self-reinforces organizations' social positions, which mostly reflect the quality of

[9] Sociologists denominate reputation as status; nonetheless, the main concepts are similar in sociology, economics and organization. In the sake of standardization, we have adopted the term "reputation" to denote both reputation and status.

interaction and the effort to prevail on other players' mind rather than the quality of organizations *per se* (Gould, 2002). Overall information inferred from organizations' reputation whether in the form of signal or endorsement is valuable for the observant party.

Even when many scholars pay attention to the determinants and process an organization follows for building reputation (i.e., Rao, 1994; Rindova et al, 2005), there are some questions still unclear. Some scholars study the effect of interaction with other players (Rao, 1994), other stream of research devote attention to efforts to keep in other people's mind (Fombrun and Shanley, 1990); and others to the quality of the organization's inputs and productive assets (Gaines-Ross, 2000; Rindova et al, 2005; and Saxton, 1998). Although theorist suggest that the quality of productive assets influences other's perception of organization reputation, (Hall, 1992, 1993; Rindova, 2005) there are few studies that find strong support. Moreover, it is difficult to measure the quality of productive assets, and to find a direct link between them and reputation, and their effects on organizations success.

We attempt to find an effect of quality of productive assets on organizations success, through a reputation building process. By analyzing over a network of more than 3000 open source software projects, the present study assess the quality of productive assets of an organization as the quality of individual programmers who belong to a project. We consider the signal of past performance and their interrelation to other projects, and we record changes in individual commitment to the project. For that purpose, we analyze projects for the development of software within the framework of open source mode of production whose members could perform programming activities simultaneously in several projects. As a signal of project's reputation due to productive assets, we consider the aggregate past performance over all the project members along the whole network. As an endorsement, social interaction or indirect dimension of reputation, we consider the network degree these projects achieve by individuals belonging simultaneously to other projects. To test the strength of the effect of building reputation by past performance and previous relations to other projects, we study changes in allocation of productive assets as the nature and the extent of programmers' commitment to the focal project for assessing in which cases the signal or endorsement prevails. The most significant difference between our study to other analyses in the field is that our setting allows others actors in the network –and even outside the network, to directly observe the quality of individual programmers: programmers create and submit their code to projects, thus other programmers can assess both output quality and programmer skills. This particular setting allows us to differentiate the self-reinforcement of reputation within the projects subject to analysis, isolating the signal and endorsement coming from network interactions. Results indicate that aggregate individual past performance and network degree increases success, that negative changes in individual commitment to the project negatively affect success, while positive changes increase success. Finally, our results show that past individual performance improves the effect of changes in individual commitment, while network degree also improves the effect of changes in individual commitment, but has an overall negative effect. In the diffusion of projects over a network, both past performance and network degree are important sources for sustaining reputation, but the effects of network degree are somewhat adverse when considering changes in individual commitment.

In the following section, this study addresses the theoretical framework and hypothesis, in section 3, the data setting, variables and regression models; in section 4 we present our results, and finally in section 5 we discuss our findings, acknowledge our limitations and postulate ideas for future research.

II. Theoretical framework, model and hypotheses.

Studies about the intangible nature of reputation and the process through which organizations achieve reputation are common in the grounds of economics, sociology, management and organization. Economic theorists devote their attention to the role of reputation for solving information asymmetry problems. Sociologists study the nature and extent of social interaction to achieve high-ranking status orders. Management and organization theorists care about the process an organization follows to build a reputation.

In information economics, the study of an actor's reputation is widely spread. The focal point is that agents care about their reputation if, when interacting with other actors, they foresee their current actions may affect future outcomes (Fudenberg and Levine, 1992; Stiglitz, 2000). This framework posits a twofold purpose for having good reputation. On one side, actors commit to develop activities or show behaviour consistently with their quality because building and maintaining a reputation is profitable in the long run (Neral and Ochs, 1992; Heugens et al., 2004; Tirole, 1996); on the other, commitment to perform such behaviour reveals a signal about the actor's quality (Schmidt, 1993). Actors invest in reputation because it leads to better market opportunities: individuals increase their value in recruitment processes; and organizations can charge higher prices or achieve higher market shares.

Within the information economics framework, reputation acts through two mechanisms; it is a signal for addressing the adverse selection problem (Schmidt, 1993; Stiglitz, 2000) and it is an incentive for addressing the moral hazard problem (DeJong et al, 1985; Fudenberg and Levine, 1992; Neral and Ochs, 1992; Stiglitz, 2000). If an actor faces an adverse selection problem about an individual agent or a firm products' quality, reputation mitigates the problem (Allen, 1984; Shapiro, 1983; Spence, 1973); reputed organizations are more likely to have a better performance in the future, as long as their signal relates to own organization's quality and implies its own effort (Spence, 1973; Tadelis, 1999). Organizations' effort involve a cost to develop the signal; this cost should be high enough that low quality organizations cannot afford it, but sufficiently low that high quality organizations are willing to incur into such cost (Spence, 1973). If the signal cost keeps within this range, high quality organizations will separate from low quality ones; if the signal cost is outside this range, high quality organizations will pool with low quality ones, (Spence,1973). At the end, actors who need to infer about organizations' quality will know that in a setting where everyone can incur in the same level of effort, reputation will no longer have a value. Therefore, reputation is a perception about organization's quality based on the signal its previous performance (Allen, 1984; Kilduff and Krackhardt, 1994; Spence, 1973; Tadelis, 1999).

Reputation is also an incentive in itself, because actors care about sustain their own reputation by constantly performing well, reducing the moral hazard problem (Stiglitz, 2000). Actors of every kind, ranging from individuals (Neral and Ochs, 1992), firms (Schmidt, 1993; Tadelis 2002) to governments (Veugelers, 1993) constantly concern about sustaining their reputation. It is rational to cooperate with other agent, or to perform high quality activities today if the discounted value of future interactions is higher (Banerjee and Duflo, 2000; Cripps et al, 2004; Fudenberg and Levine, 1992; Kim, 1996). Explicitly, organizations are willing to give up short-term gains in the pursuit of getting higher benefits in the long term, thus building their reputation, (Neral and Ochs, 1992). Therefore, organizations need to sustain behaviour in order to maintain their reputation, (Cripps et al, 2004; Fudenberg and Levine, 1992). Reputable organizations care about self-reinforcing the reputation they behold.

Sociology studies on reputation involve the interaction among organizations and other actors. In his seminal study about of reputation effects in market competition, Podolny, (1993) defined a producer status in the market as the perception of producer's products quality in relation to her competitor's products perceived quality. This definition inherently assigns to reputation the character of a hierarchy, product of a subjective judgment from other actors. In this context reputation acts through the mechanism of endorsement.

Whilst many scholars agree that subjective evaluation about quality is the main determinant of an organization's reputation (Podolny, 1993, Stewart, 2005); it is not evident the source of the quality under subjective evaluation. Gould (2002) states most studies about status divide into two main theories. One has roots in social position within a hierarchy; the other relies on market theories. Scholars that defend organization reputation as a product of social position in a hierarchy, propose that differential status emerges from the relative advantage some actors have from occupying high quality positions, regardless the preconditions that led to occupy such position (Gould, 2002). Positions are expected to be correlated to the potential of an actor's advantages over other in a broad sense, from familiar background to education. Market theories about reputation propose it as the result of a signalling game (Podolny, 1993; Gould, 2002). Reputation emerges from the information about an organization provided by her peers, judges and other stakeholders (Benjamin and Podolny, 1999; Gould, 2002; Rao, 1994; Rindova et al., 2005). In the presence of uncertainty about someone's quality, a reliable source of information is her own acquaintances (Podolny, 1993). Peers' regard works as a signal, as the actors have, until some extent control over their peers' subjective evaluation, by influencing opinion through performance. Moreover, high quality organizations incur in less costly signals than low quality organizations. Reputation differentiation is a product of differentiation in quality of the actions organizations make, (Gould, 2002). However, reputation as an incentive has a self-reinforcing character, where organizations are careful to sustain their signal by working on the same features that led to generate the signal. Besides, reputation requires the development of a high regard that lasts enough to catch others' attention; self-reinforced through social interaction (Gould, 2002; Kilduff and Krackhardt, 1994; Larson 1992; Stuart et al, 1999). For sociologists, the signal plus the bonding to others enhances an organization's reputation.

Organization and management theorists' interest in the process for building reputation relies on the value an organization obtains from it (Rindova et al,. 2005). Some researchers focus on previous and current performance as determinant of reputation. For example, previous and current performance influences market perception, increasing the opportunities to acquire higher earnings (Fombrun and Shanley, 1990); loyalty of soccer team supporters increases with results in championships (Czarnitzki and Stadtmann, 2002). Another branch of research on reputation recognizes endorsements as a source for reputation building. For example, certifications obtained by car manufacturers in quality contests improves their life expectancy in market (Rao, 1994); business schools rankings in business magazines increases their prominence in stakeholders' minds, (Rindova et al., 2005); celebrity and expert endorsements diminish the risk associated to firm's products (Biswas et al, 2006; Dean and Biswas, 2001; Seno and Lukas, 2005; Wang, 2005). In a likely way, some scholars study endorsements from social networks and alliances. Cooke and Ryan (2000) study brand strategic alliances, and find that some organizations invest in attaching their brand to other reputable brands to acquire higher market shares. Similarly, the Zhou et al, (2007) find a strong support for the idea that multinational enterprises obtain a rapid entry into market when approaching local social networks because these networks endorse organization's activities and products. Finally, a third branch of research extends on the organizations' internal characteristics to build and sustain reputation. Heugens et al, (2004) propose there are some reputation management capabilities that make some organizations more successful, when dealing with internal information and communication. Rindova et al, (2005) find that the quality of inputs is a good source of organization's reputation because they actually affect the quality of products, and propose, but does not find support, that quality of productive assets increase an organization's reputation because stakeholders can make assumptions about the quality of produce. Gaines-Ross, (2000) and Saxton (1998) propose the CEO's reputation is an important source for improved performance for a firm. Thus, for organization theorists, an organization's reputation refers to current and previous performance, social interactions, leader's reputation and internal characteristics to build reputation.

A model for reputation building.

Theorists suggest that higher reputation leads to the possibility of charging higher prices (Allen, 1984; Podolny, 1993; Rindova et al., 2005) or having higher market shares (Podolny, 1993). A higher market share comprises an increase in demand for the organization's produce. Therefore, we can expect that if organizations invest in their reputation, they will obtain higher demand for their products or will charge a higher price; therefore, they will have greater profits and become more successful. To invest in reputation, organizations should continuously develop activities that provide reliable signal overtime, and thus sustain their reputation. So far, we have discussed that, from organization theory point of view, but consistently to economics and sociology, there are three sources for building a reputation, current and previous performance, social interactions and internal characteristics. Current and previous performance implies recognition of the sources of past success and a quest for continuously improving productive processes.

We previously stated that organizational reputation relates to the information about we can gather by peers, judges and other stakeholders (Benjamin and Podolny, 1999; Gould, 2002; Rao, 1994). In addition, organizations that have prominence in stakeholders' mind enjoy higher levels of reputation, (Rindova et al., 2005).Therefore, in the quest for having prominence in stakeholders mind, an organization that produces high quality goods has interest in building links to other actors that can provide reliable information about their quality. These social interactions reinforce the organization's social position. In such a case, we can state that

Hypothesis 1: Organizations that create reputation by developing links to other organizations or agents obtain a higher success.

Good quality products require the use of with good quality inputs for their production and pass through good quality process, (Rindova et al, 2005). An organization's reputation is only as good as that of its members, where each member has individual and unique traits such as talent, diligence or honesty (Tirole, 1996). Past individual behaviour conveys information about these traits and generates individual and organizational reputations, (Tirole, 1996). A process for building reputation not only requires an investment in human, intellectual or production assets, but to be sustained by organizations that should care about every organization member becomes promoter and custodian of reputation, (Hall 1992, 1993). Organizations willing to send a signal to the market about the quality of their product would be interested in create a pool of high quality productive assets that ensures high quality goods and reinforce their reputation. Therefore, Hypothesis 2: Organizations that create reputation by developing a pool of high quality productive assets obtain higher success.

In organization are common the changes in allocation of productive assets, nonetheless these changes can affect output and thus reputation. It is the case of changes in technological processes, which can improve or decrease quality of goods produced. They can be also changes in the management team, whose decisions can improve or decrease organization's financial performance (Rao, 1994); or changes in the intellectual capital, that in the case of knowledge intensive industries, can sustain or decrease the products' quality (Powell and Snellman, 2004). Moreover, if organization changes allocation of productive assets, in such ways that some of its productive process cannot longer access to high quality productive assets, its reputation will suffer loses.

Hypothesis 3: Changes in allocation of high quality productive assets affects organization's reputation and organization success.

There is low academic consensus about what dimension – social interaction or signal in markets - a higher relative importance to solve information asymmetries has. Most empirical studies find a higher significant effect from social interaction and processes for creating reputation than the signal (Benjamin and Podolny, 1999; Kilduff and Krackhardt, 1994; Podolny, 1983; Rindova et al., 2005). A result like that might have a double interpretation: either is better to develop good relationships with other actors instead of to have good performance, or that the effect of signal is still not fully captured in research [[10]]. Our presumption is that the effect of social interaction and the signal in the market moderate the effects of changes in the allocation of productive assets, but the overall effects go in opposite direction. When organizations rely solely in social interaction and changes in their allocation of productive assets leads to less amount of devoted assets for the productive task, social interaction helps to moderate the effect of such distraction of effort; as organization is well known, these changes will not have a great impact on success. When organizations have sent a signal of high quality and changes in allocation of productive assets leads to less amount of devoted assets for the productive task, the signal would prevail and create an overall positive effect; therefore,

[10] Rindova et al. (2005) recognize this possibility in their study. They consider perceived reputation as the reflection of high quality inputs and high quality productive assets; while there can be a consensus about value of inputs, there is low consensus about quality of assets, mainly if they refer to knowledge assets.

Hypothesis 4a: Social interaction positively moderates the effect of changes in allocation of productive assets, but creates an overall negative effect.

Hypothesis 4b: Signal positively moderates the effect of changes in allocation of productive assets, and creates an overall positive effect.

III. Data and Methods

To accomplish our goal and test these hypotheses about an organization's reputation we use a database that provides details about the performance quality of individuals, their perceived quality by other actors and their effect over an organization. Moreover, the ideal data should also cover the Podolny's concerns (1993) about the gap between reputation and actual quality: time lag, stochastic process, relations within social structure and second-order nature of reputation. A social network of Open Source Projects could meet these requirements as long as it includes the nature of relationships among members, the resources provided by members, and the results from the interaction, over some period. Software creation through open mode of production involves the contributions of many individual programmers, so it is possible to measure the effect of individual contribution in overall reputation. Although Open Source Projects do not necessarily rely on contracts to hire programmers, and many of them do not seek direct economic rewards, projects constitute organizations in themselves. Open Source Projects fit in March and Simon's (1958) definition of organizations: *"systems of coordinated action among individuals and groups whose preferences, information, interests, or knowledge differ"*.

3.1. Data setting

The data we use in this analysis come from the SourceForge.net Research Data (Department of Computer Science and Engineering, University of Notre Dame). SourceForge.net is the largest repository of open source software; it hosts over 140,000 projects and gathers over 1.5 million registered users. SourceForge.net belongs to OSTG, Inc, which has shared activity data with the University of Notre Dame for the purpose of academic research on Open Source Software, under the condition of sharing the data with other researchers interested in open source software phenomenon (Christley and Madey, 2005).

Sourceforge.net as a repository has several characteristics that promote simultaneous production of code throughout a network. The purpose of the repository is to provide a platform for software development over a worldwide web infrastructure where many individuals interact for knowledge exchange. They host projects and provide tools that allow asynchronous communication, stock people's contribution, and screen software for redundant knowledge. Project creation is similar to start ups: an initiator registers the project in repository, declares project's purpose, objectives, goals, target markets, technical characteristics and team workforce and asks for *intellectual* capital venturing. Then Sourceforge.net administrators authorize the creation of a space for hosting the project if it satisfies the main premises for the repository. Sourceforge.net has some parallelisms to a marketplace where projects needing intellectual capital meet their intellectual venture capitalists; these intellectual venture capitalists are programmers outside to the project who are willing to devote their effort and knowledge to achieve project's goal. Once the repository hosts the project, it records and controls all activities. Every time there is some activity in the project, the repository electronically archives the information; these activities may include communications among members, forum posts, and more importantly, the artifacts –or modules of software code, produced by the people engaged in programming activities. As an important feature of a repository is to open the access to projects to everybody on the worldwide web, every single person, registered or not on Sourceforge.net, may see the project, download it, look at the code, look at the authors and contribute. The repository records changes in the team workforce, so it is possible to know if there are new members or if some of them have already quitted the project. As repository tracks all artifacts records, we can identify which and how often members contribute to software development into their projects and what extent they contribute to other projects. Particularly, our dataset is a sub-sample of 3196 valid observations over twelve months of projects aimed at developing game software.

To get this sample, we look at the monthly dumps of data and select all projects that belong to the category of GNU Public License (GPL) [[11]]. We restrict the sample to such a license to guarantee that projects were not subject to any restriction for copying, adoption and distribution, so all software in our sample is equally prone to reach the same audience. Then, we limit the sample to games in a general category. We pick this sub-sample because game software can achieve a mass of non-skilled persons, therefore their market is not constrained to programming skills and abilities of end users, as many other software projects are. Furthermore,

[11] GPL grants the programmers and users of software the privileges or freedom to distribute and modify copies of the software, and transfers those privileges to further developments.

we restrict the sample for the projects that were alive during the whole sample; we take this decision because dataset classifies a project as 'alive' –when the repository still hosts the project and 'dead' when they quit the repository. Unfortunately, it is impossible to distinguish if the project has quitted the repository because it went independently, or because it moved to another repository; thus, their disappearance does not necessarily relate to market demands, or product lifecycles, but to managerial premises. Finally, we look at those projects whose information on downloading activity is available. As the source of information comes from relational database, some observations could be missing in the joining process. We follow this procedure over twelve months running from February 2005 to January 2006, to build an unbalanced panel with 25491 total observations.

3.2 Variables.

In the Sourceforge.net setting, programmers freely contribute to one or several projects without receiving any compensation or salary. Software production only requires the intellectual capital of programmers. Moreover, consumers do not need to pay a price for acquiring the software, consumers only need to browse over the available software and download the one according to their preferences. Particularly, we observe that in this context, higher reputation would lead to higher market share, rather than higher prices. Therefore, we depict our variables to adjust to the setting.

Dependent Variable. Project success DOWNLOADS.

We present project success as a dependent variable; for that we use number of downloads per month as a measure of project success. A download means one user retrieves the executable files of the software for her private use; thus downloads are an output measure of success or popularity among a mass of users. Number of downloads is consistent with the nature of a software game as a digital good through Internet, because downloading is the sole way to access it; furthermore, it is a common measure of performance in OSS (Crowston et al, 2003; Grewal et al., 2006).

Independent Variables.

The demand for open source game software relates to user's individual preferences and an inference about the quality of the product. Our independent variables will attempt to cover the features we claim in the previous section as determinants of the inference of project's reputation, and include other that *somewhat control for the actual quality of the project. We can disentangle project reputation from individual reputation as follows*.

Project's social interaction (DEGREE)

We use as proxy of social interaction the network degree a project has through its members. As any single programmer could simultaneously belong to other projects, they interact with other people throughout the network. Interaction serves as a social device to increase project reputation because it increases the visibility of the project. We denote as DEGREE the number of other projects to which each project connect through its member by simultaneously belonging to them in every period, $Dj=\Sigma_i Pi$; where Pi is a project to which individual i belongs, and Dj is project J's degree.

Quality of productive assets (ASSETS QUALITY)

In previous section, we claim that organizations that develop a pool of high quality productive assets send a signal to the market about their ability to produce high quality goods. In knowledge intensive economies, organization's main productive asset is its intellectual capital (Powell and Snellman, 2004). We use as proxy of the pool of high quality of productive assets the aggregate, -current and past- individual performance on the network of all members of the projects. For each period, we measure the cumulative on t number of artifacts – pieces of code sent to solve projects' tasks-, a member i sent to any other projects K within the network and own project J, and sum over all project J's members, $\rho jt=\Sigma_i \Sigma_t \omega it$. Where ωit is an artifact sent to any project in the network by individual i in time t, and ρjt is the project *J's* aggregate individual performance. We recognize we make a strong assumption on a linear function that relates all members' contributions; nonetheless, the variable is an imperfect measure of the abilities that programmers have to detect and solve programming puzzles. We use ASSETS QUALITY as notation for Quality of productive assets.

Changes in allocation of productive assets ($\mathrm{COMMIT}_{\mathrm{N}}$)

We want to devote attention to changes in the allocation of productive assets that may affect organization's reputation. In our setting, individuals are the productive assets, and changes in their level of commitment towards the project represent a change in allocation of productive assets. In particular, we are interested in detecting whether the level of commitment of an individual within programming activities changes project success. All individuals member of a project can choose the level of engagement they commit to the project; moreover, there are no punishments or explicit pressures to adopt certain level of engagement. Members can choose between being active contributor exclusively to the project; an active contributor to the project and other projects; an active contributor to other projects; and to do not contribute in any project at all. Here contributions refer to develop programming activities or, particularly, sending artifacts. We measure how many individuals fall in each category and check for changes respect to previous period, we denote as COMMIT_N, where N is a particular case. We use COMMIT_1 when individuals move from being sole contributors to project *J* to contribute simultaneously to project *J* and any other project *K* within the network. We use COMMIT_2 when individuals move from contributing to *J* to contribute solely to *K*. We use COMMIT_3 when individuals move from contributing to any *J* or *K* to do not contribute at all. We use the change toward being exclusive programmer to *J* as the base for our regressions.

Other Contributors (OTHER CONTRIBUTORS)

We use two complementary measures to control for actual project's quality. Even when the projects create broad network for social interaction, a reliable pool of productive assets, and a constant individual commitment towards the project, just to create reputation, there should be high quality products that actually sustain their reputation. A measure of project's actual quality could be the interest it creates in other people outside the project within the network. A project receives lots of attention if other programmers start sending artifacts to its underlying code. If a project receives the attention of many other programmers outside its boundaries, then we expect it has good quality, it has interesting features, or at least relevance or value for each programmer. We denote OTHER CONTRIBUTORS number of outsider contributors participating in a given period. Outsider contributor explicitly means any single individual who sends artifacts to project *J*, when she does not appears in the front cover of the project as project member.

Artifacts' Growth (ARTIFACTS GROWTH)

It is possible to measure project quality as its efficiency in data processing. For that purpose, we look on the number of pieces of code or artifacts that a project receives in each period. Project life cycle has a close relationship to the number of artifacts received in a given period. When a project is in its early stages, it requires a strong programming activity to develop its main features. As it evolves, the requirements for artifacts diminish, if software is reliable and efficient, as it will not have programming, or running problems. Therefore, comparing two projects, each of them at the same development stage, the one that requires more artifacts is more likely to be less efficient. However, a rough measure like this could indicate that the one requiring more artifacts is more complex; therefore, we use the growth in the number of artifacts from one period to another. Thus, keeping stage constant, projects whose number of artifacts grows from period to period, have poor quality, as they need a lot of code to perform their activities. We denote ARTIFACTS GROWTH the difference among periods of the number of artifacts received by a project.

Control variables

In the present study, we have included the project's characteristics as control variables. On one hand, these control variables reflect the characteristics, which may induce differences in dependent variable because of demographic issues, such as size and stage of a project; but also those referring to their technical peculiarities, which could make it more attractive for both programmers and users. Our control variables include:

Project size (SIZE)

We consider as project size the number of members of project J, for accounting for all variance due to a higher number of people involved in the project.

Project stage (STAGE)

OSS projects included in our database are at different stages of development, going from planning, pre-alpha, alpha, beta, production, mature and inactive. The stage of the project and downloads are strongly associated. At its first stages, the core tasks relate to the creation of an initial system that will evolve over time, thus less demanded by users. At the last stages, the core task is the diffusion of the product, thus highly demanded by users. We controlled for the evolution as the project may require different knowledge along its stages.

Characteristics

We controlled for characteristics such as programming language, operating system, user interface, intended audience, and speaking language of the game.

Previous project success:

We control for the number of downloads at the previous period to take into account the possibility that project's success in past period attract the attention of other users. We use the lagged dependent variable.

3.3 Method of analysis.

In order to test our hypotheses on a dynamic approach, we use a differences-in-differences fixed effects estimator; that is taking differences in both dependent and independent variables on a monthly basis, except for the $COMMIT_N$ and ARTIFACTS GROWTH variables, that were already in that form. Therefore, we condition a variation in success at month *t+1* with respect to month *t*, to a variation on reputation measures and controls at month *t+1*with respect to month *t*, controlling for the project's fixed effects over the period of study. We want to measure how variation on reputation influences the monthly rate of downloading. The use of this type of regression method obeys our intention of testing how differential rates of reputation building strategies condition project success. The fixed effect estimator allows us to isolate the effect of a particular project, whose characteristics make it more attractive for programmers. The differences-in-differences fixed effects estimator allows us to isolate variations in dependent variable due to changes in independent variables conditional to the particular project's characteristics, which make them more attractive for programmers. This approach is a consistent transformation for Pareto distributions, as is the case of our dependent variable. Figure 1a, shows the distribution of variable DOWNLOADS, while Figure 1a shows the distribution of the differences in differences transformation of DOWNLOADS. A Pareto distribution is the graphical representation of what Merton (1968) defined as the "Matthew effect": those who already have will be given more. In this setting, very few projects are profusely downloaded; meanwhile many others never attract any user.

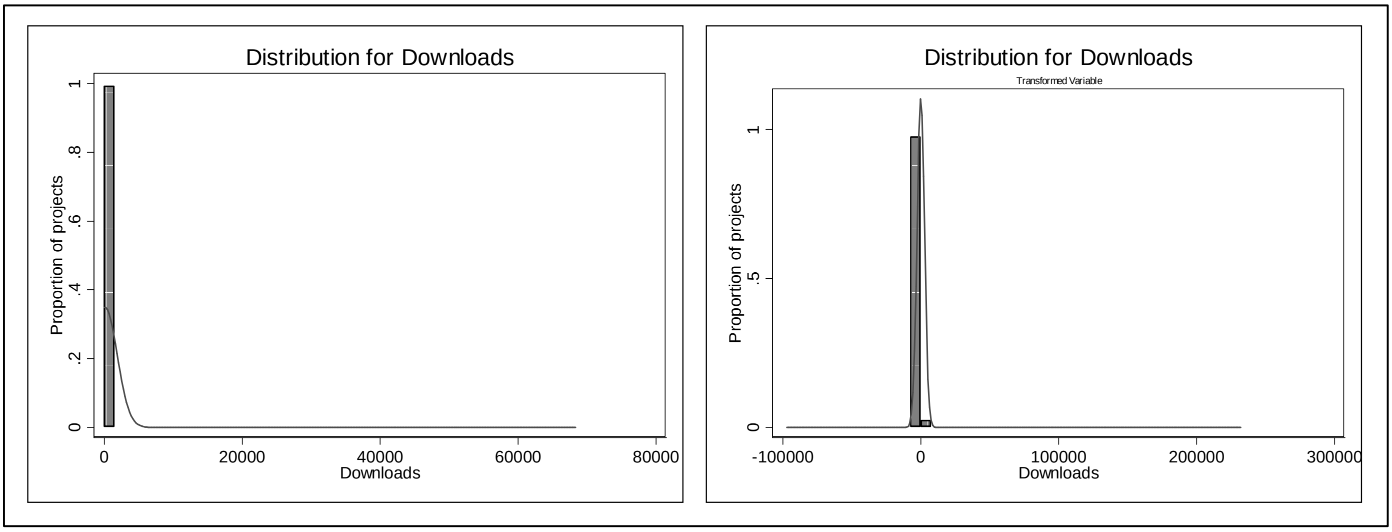

Figure 1. Distributions of monthly downloads, original and transformed variable.

IV. Results.

Table 1 shows main descriptive statistics and Table 2 presents the correlation matrix for the main variables included in our analyses. Table 3 shows the results of differences-in differences fixed effects estimators. Model 1 presents our base analysis, where variations in DOWNLOADS in previous period has a negative and significant effect in variations in DOWNLOADS in current period; this result appears also in all other regressions depicting a pattern common in experience and cultural goods, that earn most of their revenues when they entry on the market, but decrease over time. The significance and negative effect of the artifacts' growth (ARTIFACTS GROWTH) denotes that projects that from period to period receive higher amounts of artifacts become less downloaded than those who receive less amounts of artifacts. This result reflects that keeping everything else constant, projects that still need to solve programming issues have low quality, as we proposed earlier. Variable's coefficients values, signs, and significance remain consistent and stable throughout all regressions.

Statistics	Mean	Max	Min	Range	Std. Dev.	N
Δ DEGREE	-0.01	39	-34	73	0.94	32263
Δ ASSETS QUALITY	-0.37	158	-531	689	6.25	32263
$COMMIT_1$	-0.01	2	-8	10	0.14	35948
$COMMIT_2$	-0.03	3	-11	14	0.26	35948
$COMMIT_3$	0.04	14	-3	17	0.34	35948
$COMMIT_1$ X $DEGREE_{t-1}$	-0.05	9	-224	233	2.01	35948
$COMMIT_2$ X $DEGREE_{t-1}$	-0.08	114	-385	499	3.02	35948
$COMMIT_3$ X $DEGREE_{t-1}$	0.17	392	-114	506	4.46	35948
$COMMIT_1$ X ASSETS $QUALITY_{t-1}$	-0.26	618	-3632	4250	21.46	35948
$COMMIT_2$ X ASSETS $QUALITY_{t-1}$	-0.16	494	-531	1025	8.79	35948
$COMMIT_3$ X ASSETS $QUALITY_{t-1}$	0.57	6356	-370	6726	35.80	35948
Δ DOWNLOADS	18.15	232030	-97049	329079	2699.18	28802
Δ SIZE	0.02	24	-12	36	0.49	32263
Δ OTHER CONTRIBUTORS	-0.03	357	-357	714	3.46	32263
ARTIFACTS GROWTH	-0.69	1000	-3276	4276	30.97	32263
Δ STAGE	0.00	0.83	-0.83	1.67	0.02	32263

Table 1. Descriptive statistics.

correlations	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1 *Δ DOWNLOADS*	1																
2 *Δ DOWNLOADS* $_{t-1}$	-0.541 *	1															
3 *Δ SIZE*	0.016 *	0.008	1														
4 *Δ OTHER CONTRIBUTORS*	-0.006	-0.003	-0.182 *	1													
5 *ARTIFACTS GROWTH*	-0.049 *	-0.110 *	-0.114 *	0.330 *	1												
6 *Δ DEGREE*	0.005	-0.004	0.489 *	0.004	0.102 *	1											
7 *Δ ASSETS QUALITY*	-0.017 *	-0.057 *	-0.059 *	0.136 *	0.385 *	0.153 *	1										
8 *COMMIT* $_1$	-0.018 *	-0.033 *	-0.472 *	0.202 *	0.412 *	-0.098 *	0.373 *	1									
9 *COMMIT* $_2$	-0.006	-0.006	-0.028 *	-0.020 *	0.006	0.175 *	0.219 *	0.000	1								
10 *COMMIT* $_3$	0.019 *	0.018 *	0.399 *	-0.110 *	-0.262 *	-0.028 *	-0.360 *	-0.510 *	-0.764 *	1							
11 *COMMIT* $_1$ *X DEGREE* $_{t-1}$	-0.003	-0.022 *	-0.585 *	0.109 *	0.279 *	-0.241 *	0.281 *	0.802 *	0.010	-0.443 *	1						
12 *COMMIT* $_2$ *X DEGREE* $_{t-1}$	-0.003	-0.005	-0.038 *	-0.020 *	0.002	0.200 *	0.168 *	0.014 *	0.552 *	-0.430 *	0.013 *	1					
13 *COMMIT* $_3$ *X DEGREE* $_{t-1}$	0.004	0.014 *	0.479 *	-0.064 *	-0.195 *	0.053 *	-0.286 *	-0.500 *	-0.380 *	0.663 *	-0.653 *	-0.689 *	1				
14 *COMMIT* $_1$ *X ASSETS QUALITY* $_{t-1}$	0.012 *	-0.024 *	-0.340 *	0.149 *	0.262 *	-0.119 *	0.287 *	0.566 *	-0.001	-0.320 *	0.725 *	0.009	-0.520 *	1			
15 *COMMIT* $_2$ *X ASSETS QUALITY* $_{t-1}$	0.023 *	-0.021 *	-0.041 *	-0.085 *	0.003	0.115 *	0.163 *	0.001	0.395 *	-0.305 *	0.031 *	0.575 *	-0.415 *	-0.034 *	1		
16 *COMMIT* $_3$ *X ASSETS QUALITY* $_{t-1}$	-0.013 *	0.020 *	0.384 *	-0.102 *	-0.188 *	0.118 *	-0.252 *	-0.469 *	-0.108 *	0.414 *	-0.676 *	-0.167 *	0.660 *	-0.913 *	-0.254 *	1	
17 *Δ STAGE*	0.017 *	-0.021 *	-0.028 *	0.001	0.001	-0.016 *	0.005	0.074 *	-0.009	-0.022 *	0.064 *	-0.003	-0.027 *	0.093 *	0.000	-0.054 *	1

* significant at 5%

Table 2. Correlation matrix

In Model 2, we have included variation in DEGREE and variation in ASSETS QUALITY as explanatory variables. Variation in DEGREE is positive and significant, explicitly when projects have increase in one more link their network degree then receive 105 more downloads. This result supports our Hypothesis 1, where projects that spread their reputation through linking to other projects receive more downloads. Variation in ASSETS QUALITY is also positive and significant. It means if at least one of project members sends one more artifact; the project receives 85 more downloads. Assets quality is the cumulative number of artifacts that members of a project send to own project and other projects; therefore, we might interpret changes in this variable as a signal of the quality of the pool of productive assets, satisfying our Hypothesis 2.

Model 3 presents results where we have included those changes in allocation of productive assets, and in our setting, changes on commitment to the development of activities for project *J*; $COMMIT_1$ is positive and significant, while $COMMIT_3$ is also significant but negative. With $COMMIT_1$ we found that when an individual moves from solely developing code to project *J* towards developing code also for project *K*, then project *J* receives around 612 more downloads respect to previous period. This particular change on commitment implies an activity that reinforces reputation by providing individual problem solving abilities to other projects, and creating a link through contribution to other projects. On the other hand, $COMMIT_3$ resulted negative although significant at 10%. Thus when an individual moves from developing code for other projects *K*, and suddenly stop contributing to them, project *J*'s success decreases in 344 downloads; it appears to be negatively regarded by peers, and could mean that when members project *J* stop submitting artifacts to other projects *K* then project *J* loses visibility. This joint result supports Hypothesis 3. In Model 3, we also observe that increases in size have a positive significant effect.

Model 4 includes variation in DEGREE, variation in ASSETS QUALITY and changes in commitment. DEGREE and ASSETS QUALITY still present positive, significant and similar effects as in Model 2. Nonetheless, an organizational change in commitment ($COMMIT_2$) is the only variable that resulted negative and significant at 10%. This result implies that when an individual changes from sending artifacts to both project *J* and other projects *K*, to send only to projects *K*, project *J* decreases in 386 downloads. These results also provide complementary support to our Hypothesis 3.

Dependent variable: Δ Project Success

	(1)	(2)	(3)	(4)	(5)
Δ DEGREE		105.750 ***		106.253 ***	
		(21.950)		(21.986)	
Δ ASSETS QUALITY		85.230 ***		89.668 ***	
		(8.170)		(9.116)	
$COMMIT_1$			612.020 **	-356.243	
			(297.560)	(315.076)	
$COMMIT_2$			-243.150	-386.961 *	
			(210.170)	(212.143)	
$COMMIT_3$			-344.790 *	-155.855	
			(196.040)	(198.053)	
$COMMIT_1$ X $DEGREE_{t-1}$					-59.682
					(38.611)
$COMMIT_2$ X $DEGREE_{t-1}$					-98.388 ***
					(25.443)
$COMMIT_3$ X $DEGREE_{t-1}$					-87.310 ***
					(21.207)
$COMMIT_1$ X ASSETS $QUALITY_{t-1}$					11.426 **
					(3.842)
$COMMIT_2$ X ASSETS $QUALITY_{t-1}$					8.123 **
					(3.089)
$COMMIT_3$ X ASSETS $QUALITY_{t-1}$					7.529 ***
					(2.287)
Δ $DOWNLOADS_{t-1}$	-0.676 ***	-0.674 ***	-0.675 ***	-0.674 ***	-0.675 ***
	(0.006)	(0.006)	(0.006)	(0.006)	(0.006)
Δ SIZE	17.260	36.300	224.080 ***	44.605	236.068 ***
	(43.920)	(50.070)	(66.220)	(70.442)	(62.823)
Δ OTHER CONTRIBUTORS	3.410	6.310	6.559	5.652	11.263 **
	(4.790)	(4.800)	(4.850)	(4.871)	(5.370)
ARTIFACTS GROWTH	-38.760 ***	-45.110 ***	-39.150 ***	-45.854 ***	-39.581 ***
	(1.530)	(1.619)	(1.540)	(1.662)	(1.583)
Δ STAGE	1174.090	1103.910	455.330	1303.345	188.945
	(1113.530)	(1110.150)	(1133.000)	(1133.220)	(1158.345)
N	25491	25491	25491	25491	25491
F	679.900	588.730	579.590	512.330	447.510
p>F	0.000	0.000	0.000	0.000	0.000
rho	0.308	0.313	0.310	0.313	0.310

Standard errors in parenthesis. * sig at 10%; ** sig at 5%; *** sig at 1%

Table 3. Differences-in-differences Fixed Effects Regression Estimates.

In Model 5, we included interactions between changes in commitment, and degree in previous period, and changes in commitment and assets quality in previous period. Interactions between changes in commitment and degree in previous periods means that we considered the degree project *J* has in period *t-1* and multiply by the change in commitment N, therefore its meaning refers to what extent changes in commitment affect downloads given the previous period degree of project *J*. When referring to the interactions between commitment and degree, we observe that $COMMIT_2$ X $DEGREE_{t-1}$ has a negative and significant coefficient; therefore, if a member of the project *J* changes to solely submitting artifacts to project *K*, given that in previous period had a higher degree respect to other projects, project *J* loses 98 downloads. Downloads are decreasing on previous period degree multiplied by a change in commitment. We obtain a similar coefficient to $COMMIT_3$ X $DEGREE_{t-1}$, where members stop sending artifacts completely, but project *J* had a certain degree in previous period. As the links measured by DEGREE refer only to membership and not contribution, the bonding is not tight; therefore, this result may imply that other people in the network would not be interested in *J*, given that their members, usually belonging to many other projects do not perform programming activities. This result supports Hypothesis 4a. Interactions between changes in commitment and ASSETS QUALITY in previous period ($COMMIT_N$ X ASSETS $QUALITY_{t-1}$) mean that we multiply the aggregate assets quality in previous period by the change in commitment N; therefore, it implies to what extent changes in commitment affect changes in downloads given members of project *J* achieved some aggregate quality. All three variables have positive and significant effects, whereas $COMMIT_1$ X ASSETS $QUALITY_{t-1}$ has a higher effect than other two. These results suggest that project's reputation remains even when individuals change their commitment; in any case, it is more important when individuals contribute simultaneously to project *J* and other projects *K*. These results support our Hypothesis 4b. Overall, our results present in a non-structured manner that if organizations engage in activities to sustain their reputation obtain greater success, providing support for our whole model.

V. Conclusions, limitations and questions for future research.

Creating and sustaining a reputation are not easy tasks for organizations. These processes require that organizations devote efforts to produce high quality goods, to create a signal that distinguish them from the rest of organizations in their market, to develop relations to stakeholders who can endorse them, and to be aware that subtle changes in allocation of productive assets can affect their final reputation. In this study, we find a positive effect of creating links between projects through shared membership. Members of a project that also belong to other projects help to spread the knowledge about the project and increase its demand. Strength of our study is that we can actually measure how projects interrelate through their members; therefore, our results are quite robust. Thus, the social interaction increases demand by increasing of reputation.

Organizations trying to differentiate from others through investment in high quality of productive assets will find their reward. In our particular setting, we can measure individual performance inside and outside the project, and finally aggregate over all members of the project. Project members' abilities and skills for solving tasks even for other projects provide a good reference of the quality of project's human capital. Besides, other people within the network can be aware about these abilities and skills.

Allocation of productive assets has been always an important issue in management. We go further in this study by claiming that changes in this allocation could also affect organization's reputation. Moreover, we find that distracting productive assets from production process affects final success. When an individual changes her commitment towards the project, and starts sending artifacts also to other projects, or resigns – in terms of activity, not in terms of membership- to project, then project will suffer important loses in terms of downloads. Our result is comparable to demand decrease due to changes in productive assets. These loses could be not that important if the organization procures to have strong social interaction among other projects, but still have a negative effect. Nonetheless, our study underlies the relevance of creating reputation through social interaction in order to diminish adverse effects on demand due to changes in productive assets. If organizations create reputation through investing in high quality productive assets, the story has a nicer end; changes in allocation of productive assets have a positive effect if the quality of productive assets increases. Signal from aggregate individual past performance will prevails even when some individuals change their commitment towards the project. Nonetheless, the effect will increase as the change goes from contributing exclusively to the project to contributing to also several projects. If there is a change to not contributing, or contributing to other projects, the signal also prevails, although its effect is lower.

In addition, we should remark, we included some proxies for actual quality of the project, the number of outsider contributions and artifacts growth. These controls allow us to identify the effect of reputation, keeping constant the quality of the project.

Nonetheless, our study presents some limitations. We used the number of projects to which project J connects as a measure of social interaction. However, we cannot actually measure the quality of social interaction. Most studies in reputation through social interaction acknowledge that both, quality of the relationship, and quality of the partner in relationship will affect the level of reputation. Therefore, our measure lacks these two properties, and assigns equal weight to all of these relationships. Nonetheless, we find a positive effect, which can indicate the quality of good relationships prevails of bad relationships. Another limitation is that the use of an aggregate measure of individual performance as quality of productive assets does not differentiate between the number of contributions and the actual quality of contributions; if we can weight the number by the quality then the proxy for reputation would be more accurate. In this sense, it would be interesting to measure the effect of disaggregate individual performance, because some particular individual could be leading project's quality and utterly, project's reputation. Although organizations' reputation is somewhat a complex task, recognizing its processes and mechanisms helps to develop a strategy for its creation and reinforcement over time.

VI. Bibliography.

Allen, Franklin (1984). Reputation and Product Quality. *The Rand Journal of Economics*, 15(3), 311-327.

Banerjee, Abhijit V, Esther Duflo. (2000). Reputation effects and the limits of contracting: A study of the Indian software industry. *The Quarterly Journal of Economics*, 115(3), 989-1017.

Benjamin, Beth A, Joel M Podolny. (1999). Status, quality, and social order in the California wine industry. *Administrative Science Quarterly*, 44(3), 563-589.

Biswas, Dipayan, Abhijit Biswas, Neel Das. (2006). The Differential Effects of Celebrity and Expert Endorsements on Consumer Risk Perceptions. *Journal of Advertising*, 35(2), 17-31.

Christley, S. and Madey, G, 2005, Collection of Activity Data for SourceForge Projects, *Technical Report: TR-2005-15*, Dept. of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN.

Cooke, Sinead, Paul Ryan. (2000). Brand alliances: From reputation endorsement to collaboration on core competencies. *Irish Marketing Review*, 13(2), 36-41.

Cripps, Martin W George J Mailath, Larry Samuelson. (2004). Imperfect Monitoring and Impermanent Reputations. *Econometrica*, 72(2), 407-432.

Crowston, K., Annabi, H, Howison, J., 2003, *"Defining Open Source Software Project Success"* 24th International Conference on Information Systems, http://opensource.mit.edu/papers/crowstonannabihowison.pdf

Czarnitzki, Dirk, Georg Stadtmann. (2002). Uncertainty of outcome versus reputation: Empirical evidence for the First German Football Division. *Empirical Economics*, 27(1), 101-112.

Dean, Dwane Hal, Abhijit Biswas. (2001). Third-party organization endorsement of products: An advertising cue affecting consumer prepurchase evaluation of goods and services. *Journal of Advertising*, 30(4), 41-57.

Fombrun, Charles, Shanley, Mark. (1990). What's in a Name? Reputation Building and Corporate Strategy. *Academy of Management Journal*, 33(2), 233.

Fudenberg, Drew, Levine, David K. (1992). Maintaining a Reputation when Strategies Are Imperfectly Observed. *The Review of Economic Studies*, 59(200), 561.

Gaines-Ross, L (2000). CEO Reputation: A Key Factor in Shareholder Value. *Corporate Reputation Review*, 3(4), 366-370.

Gould, Roger V (2002). The origins of status hierarchies: A formal theory and empirical test. *The American Journal of Sociology*, 107(5), 1143-1178.

Grewal, R., G.L. Lilien and G. Mallapragada, 2006, Location, Location, Location: How Network Embeddedness Affects Project Success in Open Source Systems, *Management Science*, 52 (7): 1043-1056

Hall, Richard (1992). The Strategic Analysis of Intangible Resources. *Strategic Management Journal*, 13(2), 135-144.

Hall, Richard (1993). A framework linking intangible resources and capabilities to sustainable competitive advantage. *Strategic Management Journal*, 14(8), 607-618.

Heugens, Pursey P M A R, Cees B M van Riel, Frans A J van den Bosch. (2004). Reputation Management Capabilities as Decision Rules. *The Journal of Management Studies*, 41(8), 1349-1377.

DeJong, Douglas V., Forsythe, Robert, Lundholm, Russell J., Uecker, Wilfred C., Young, S. Mark (1985). A Laboratory Investigation of the Moral Hazard Problem in an Agency Relationship/Discussion. *Journal of Accounting Research*, 23, 81-120.

Kilduff, Martin, Krackhardt, David. (1994). Bringing the individual back in: A structural analysis of the internal market for reputation in organizations. *Academy of Management Journal*, 37(1), 87.

Kim, Jeong-Yoo (1996). Cheap talk and reputation in repeated pretrial negotiation. *The Rand Journal of Economics*, 27(4), 787-802.

Larson, Andrea (1992). Network Dyads in Entrepreneurial Settings: A Study of the Governance of Exchange Relationships. *Administrative Science Quarterly*, 37(1), 76-104

Merton, Robert K. (1968) The Matthew Effect in Science. *Science*, 159(3810): 56-63.

Neral, John, Ochs, Jack. (1992). The Sequential Equilibrium Theory of Reputation Building: A Further Test. *Econometrica*, 60(5), 1151.

Podolny, Joel M (1993). A status-based model of market competition. *The American Journal of Sociology*, 98(4), 829.

Powell, Walter W, Kaisa Snellman. (2004). The Knowledge Economy. *Annual Review of Sociology* 30, (January 1): 199-220.

Rao, Hayagreeva (1994). The social construction of reputation: Certification contests, legitimation, and the survival of organizations in the American automobile industry: 1895-1912. *Strategic Management Journal:* Special Issue, 15, 29.

Rindova, Violina, Ian O Williamson, Antoaneta P Petkova, Joy Marie Sever. (2005). Being Good or Being Known: An Empirical Examination of the Dimensions, Antecedents, and Consequences of Organizational Reputation. *Academy of Management Journal*, 48(6), 1033-1049.

Saxton, M K (1998). Where do Reputations Come From? *Corporate Reputation Review*, 1(4), 393-399.

Schmidt, Klaus M (1993). Commitment through incomplete information in a simple repeated bargaining game. *Journal of Economic Theory*, 60(1), 114.

Seno, Diana, Bryan A. Lukas. (2007). The equity effect of product endorsement by celebrities: A conceptual framework from a co-branding perspective. *European Journal of Marketing*, 41(1/2), 121-134.

Shapiro, Carl (1983). Optimal Pricing of Experience Goods. *The Rand Journal of Economics*, 14(2), 497-507.

Spence, Michael (1973). Job Market Signaling. *The Quarterly Journal of Economics*, 87(3), 355-374.

Stiglitz, Joseph E. (2000). The contributions of the economics of information to twentieth century economics. *The Quarterly Journal of Economics*, 115(4), 1441-1478.

Stewart, Daniel (2005). Social Status in an Open-Source Community. *American Sociological Review*, 70(5), 823-842.

Stuart, Toby E., Ha Hoang, Ralph C Hybels. (1999). Interorganizational endorsements and the performance of entrepreneurial ventures. *Administrative Science Quarterly*, 44(2), 315-349.

Tadelis, Steven (1999). What's in a name? Reputation as a tradeable asset. *The American Economic Review*, 89(3), 548-563.

Tadelis, Steven (2002). The market for reputations as an incentive mechanism. *The Journal of Political Economy*, 110(4),854-882.

Tirole, Jean (1996). A theory of collective reputations (with applications to the persistence of corruption and to firm quality). *The Review of Economic Studies*, 63(214), 1-22.

Veugelers, Reinhilde (1993). Reputation as a mechanism alleviating opportunistic host government behavior against MNEs. *The Journal of Industrial Economics*, 41(1), 1-17.

Wang, Alex (2005). The Effects of Expert and Consumer Endorsements on Audience Response. *Journal of Advertising Research*, 45(4), 402-412.

Zhou, Lianxi, Wei-ping Wu, Xueming Luo. (2007). Internationalization and the performance of born-global SMEs: the mediating role of social networks. *Journal of International Business Studies*, 38(4), 673-690.

Conclusions.

In the first chapter, the core of our contribution relies on the measurement and identification of the collaborative structure – by identifying programmers' roles and affiliations - for knowledge sharing and the differential effects on project success related to knowledge reception and provision. As our own results show, this measurement supports the generalised reciprocity exchange theory; therefore, with the aim of finding significant sources of social capital, the necessity emerges for accounting differences in investment strategies or knowledge exchanges that complement the stock of capital.

In the second chapter, the main contribution relies in finding the effects of some determinants for contribution. One of the main concepts is the effect of individual signalling as soon as they enter into the network by contributing to the first project they subscribe. Other important concept is complementarities, as the willingness to contribute is increasing on the proportion of the categories covered by others in every project.

In the third chapter, the main contribution relies in finding how changes in allocation of productive assets can lead to decrease in success, but degree and quality of assets help to diminish the perverse effect of these changes.

Research in open source software helps to solve many issues addressed in management, and the benefits will increase as soon as information becomes richer.